

インターネットの
利用と仕組み

8

吉村 伸

Gopher、WWW、WAIS

はじめに

現在のインターネットのトラフィックでもっとも大きな割合を占めるのは ftp-data です。表 1 は、1993 年 9 月の NSFNET バックボーンのポート別のトラフィック量（上位 20 位まで）です¹。右側のバイト数でのデータの上位から、ftp-data (1)、nntp (2)、gopher (7)、www (10) の 4 つを加算すると 50% を超えます。

これがインターネットを特徴づける傾向です。ftp-data のなかにはむろん個々のファイル転送も含まれるでしょうが、抜きん出ているのがいうまでもなく anonymous FTP です。nntp は 1993 年 10 月号でとりあげた Net-News です。そして、gopher、www が今回からとりあげるアプリケーションです。今回は、初めにインターネットの現状と、起こりつつある問題点を、次にこれらのアプリケーションをカタログ的に紹介します。

インターネットのコミュニケーション・モデル

上位を占めるこれらのトラフィックは、電子メールのような特定のユーザー間の通信ではありません。データなど皆が見てよい、共有のためのトラフィックです。公開された文書、データなどといった皆が共有する資産です。telnet には、公開された図書館のデータベースを利用するためのものも含まれていますし、電子メールのなかにもメールサーバーなどからソフトウェアや文書を入手するためのものが含まれています。公開されたメーリングリストの数も多いはずで

比較的上位にある irc (Internet Relay Chat) はいわゆる“チャット”システムで、世界中の人がコンピュータを使ってキーボードを叩いてお話をしています。文字だけのコミュニケーションでこれだけのトラフィックを生ずるのは相当多くのユーザーに利用されていることが分かります。しかし、これも、1 対 1 のコミュニケーションではなくグループ・コミュニケーションの形態です。

このようにみえてくると、インターネットでは、皆で、あるいはグループで共有するためのトラフィックが圧倒的に多いことが分かります。

インターネットの構築コスト

一方、インターネットを構成する素材のコストに目を向けると、別の側面がみえてきます。

現在のインターネットは広域接続をおこなう部分では、一般に NTT などの第一種通信事業者の提供するサービスを利用しています。このため、国内インターネットでは回線費がコストのほとんどを占めています。バックボーンでは回線を束ねることでコストダウンを図ることが可能ですが²、そこからユーザーまでのアクセスに関しては、現在の専用線というシステムを使うかぎり、ほかの機器のコストにくらべてはるかに高く、諸物価と比較してもほかのものを凌駕しています。これは歴史的経緯によるところが大であり現実とはかならずしも一致していません³。

遠近格差はかつての長距離バックボーンを敷設するのに多大なコストを必要とした時代の遺物ですが、回線速度は電話何本ぶんという電話との比較にもとづいています。

1 バケットカウント 6 位の ip は、IP over IP のトラフィック、おもに MBONE (マルチキャストによる、オーディオ、ビデオ・カンファレンスなど) のトラフィックであるとみなしてよいでしょう。

2 長距離の回線費が高いためにインターネットが回線のリセール業とみられがちなのは嘆かわしいことです。

3 ここには最近話題の許認可と、規制緩和というキーワードが関係していますね。



表1 1993年9月のNSFNETバックボーン・トラフィックのプロトコル別統計

サービス名	ポート数	順位	バケット数合計	割合	順位	総バイト数	割合
ftp-data	20	1	9,604,446,650	22.192	1	3,415,293,403,250	41.304
telnet	23	2	7,148,082,700	16.517	5	494,858,358,300	5.985
nntp	119	3	3,823,828,500	8.835	2	824,291,516,050	9.969
smtp	25	4	3,194,735,750	7.382	4	534,805,452,800	6.468
domain	53	5	2,519,422,500	5.821	6	236,157,931,350	2.856
ip	-4	6	1,935,750,400	4.473	3	563,742,526,200	6.818
irc	6,667	7	1,032,263,350	2.385	9	103,426,953,150	1.251
ftp	21	8	820,992,900	1.897	11	75,192,359,350	0.909
icmp	-1	9	798,126,000	1.844	12	67,661,349,150	0.818
gopher	70	10	762,736,250	1.762	7	198,095,673,750	2.396
X0	6,000	11	375,399,850	0.867	13	64,225,062,300	0.777
vmnet	175	12	367,187,350	0.848	8	133,415,741,500	1.614
login/who	513	13	309,834,900	0.716	14	31,232,286,400	0.378
talk	517	14	271,047,850	0.626	16	27,961,837,450	0.338
(unknown)	1,023	15	264,813,950	0.612	15	29,978,357,300	0.363
www	80	16	214,267,950	0.495	10	75,401,057,950	0.912
finger	79	17	192,690,150	0.445	19	17,414,834,450	0.211
snmp	161	18	157,311,950	0.363	20	16,919,897,250	0.205
(unknown)	1,022	19	135,409,700	0.313	17	22,599,408,300	0.273
ntp	123	20	135,249,750	0.313	26	10,293,204,700	0.124

この辺りの事情は、電話技術を基礎にしていることからきています。電話という技術は、1対1の通信をおこなうことが基本です。交換機の技術は1対1の通信をやりとりするためのものであり、データの共有やグループを構成することを想定して作られてはいません。

しかし、インターネットのアプリケーションによる回線の使い方はこれとは異なっています。電話網の交換機が担っている、多数のなかから1対1のペアを選び出す構造を必要としません。現在の電話では、コストは発呼によって交換機が働く部分に集中しており、それにくらべてデータを伝送する部分はさほど大きくはありません。

共有型コミュニケーションと、プライベート・コミュニケーション

インターネットにおけるコミュニケーションの支配的なトラフィックは共有型のものです。大容量ではあるが内容の秘匿性は薄く、伝送路の信頼性は高いに越したことはありません。しかし、それにも増して、安価で大量の通信ができることが優先します。その極端な例を放送に求めることができます。このように、インターネットには放送型の要素もあります。

しかし、インターネットに特徴的なこととして、特定サービス提供者に対しての利用者が存在するわけではなく、

小さなサービス提供者が多数散在して、利用者と提供者の差が小さいことも挙げられます。

情報の即時性という点では、情報の発信源から提供されるほうが有利なのは当然です。ところが、利用者の要求の多様化にともない、より細かな専門知識で量が少ないケースも出てきます。そのような場合には、情報の海のなかに沈めてしまわずに、分野別に1カ所に集中させて知識を追究できるようにしておくほうが合理的です。広範囲に品物を揃えているデパートと専門店の差ですが、これは両方必要なファンクションでしょう。

この形態の対極にあるのが、プライベート・コミュニケーションです。これには電子メールも一部含まれますが、トラフィックが共有型のアプリケーションにくらべて低く、伝送路の信頼性が優先されるアプリケーションです。

これは、いわば電話型のアプリケーションで、その仕組みも現在の料金体系とかけ離れたものではありません。しかし前者は、従来の料金体系には馴染まない要素をたくさんもっています。現在のインターネットは、どちらかといえば共有型のコミュニケーション・モデルを可能にするように組み立てられています。本来はまったくモデルの異なるシステムを、既存の電話網の技術を利用して構築せざるをえないことが、現在のインターネットが直面しているコス

図1 MS-Windows 上の Gopher クライアント

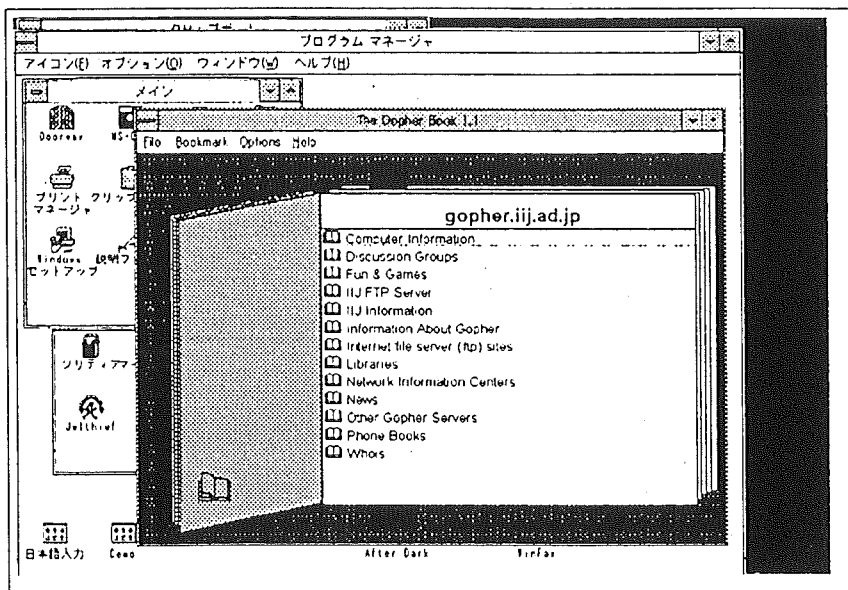
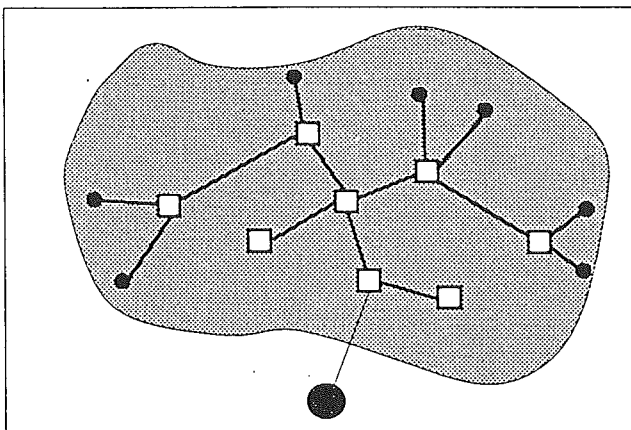


図2 インターネットのコミュニケーション・モデル



ト上の悩みの1つでしょう。

インターネットの利用を概念化すると、図2のようになります。任意の地点からどこか他の場所にある情報に接触をします。ユーザーにとっては、その情報との距離、伝達速度、応答時間などを意識することはほとんどありません。これは、wwwやGopherのように使いやすいアプリケーションのときほど顕著になります。

もちろん実際には、線をたどって特定のサーバーと通信をしています。たまたまそうになっているだけにすぎません。

このようなシステムであえて課金を考えるとすると、手許の線の太さでやりとりできる情報量くらいでしょう。

はたして、これを現実のシステムとして作れるのでしょうか？ 遠近格差は、1対1の通信ならば受益者負担の原

則を適用できます。しかし共有型では利益を得るものが分からないので、この部分が最大のネックになるでしょう。これを解消できる新しい技術の登場が期待されます。広域ATMのサービスが順調に提供されればかなり現実性があるかもしれません。

時間と速度はかろうじて意味をもちます。しかし、この傾斜があまりきついとマルチメディア化の障害になるのは明白であり、だれにとってもあまり得策ではなさそうです。

どのような技術ができて、手許まで線を引くことが最後に残ります。ここをうまく解決する方法が技術的にも、制度的にも望まれます。この点が最後に残された鍵になるでしょう。

近ごろインターネットで流行るもの

さて、近ごろインターネットでユーザーが爆発的に増加しているものの代表で、この見出しがぴったり当てはまるのが、Gopher、WWW、WAISです。そしてこれらは共有型アプリケーションとして、情報の存在する場所を意識させることの非常に少ないものの典型でもあります。

Gopher

Gopherはメニュー形式でインターネット上に散在するリソースを眺めていくシステムです。これが普及した原

因は、ユーザー・フレンドリーなインターフェイスの考え方にあると思われます。Gopher の歴史は、Minnesota 大学でキャンパス情報を提供するために作られたことに始まります。

メニューは、そのマシン内の情報であることや、他のプログラムを起動するものであること、あるいはほかの Gopher サーバーへのポインタであったりもします。Gopher の使いやすさはメニューの設計に依存しますが、1つのインターフェイスから多様なリソースへアクセスできるという仕組みだけで、こんなにも使いやすくなるということを実感させます。とくに X ウィンドウのように GUI を利用したクライアントの場合には、その威力は絶大です。

Gopher には X (図 3) だけでなく、MS-Windows (図 1) や、Macintosh 用のクライアントもあります。インターネットのリソースへアクセスするためには UNIX を知る必要があるという話をよく耳にしますが、決してそうではありません。

MS-Windows 用のクライアント (Gopher Book)⁴ は、Winsock 上で動作しています。Winsock は、Microsoft Windows 用の API で、TCP/IP を扱う共通のインターフェイスを提供しています。今回は、NetManager の Chameleon に含まれている Winsock を使って動作させました。

Gopher Book は、CNIDR (Clearinghouse for Networked Information Discovery and Retrieval) という組織が作っています。これは、NSF (National Science Foundation) と、MCNC⁵ Center for communication の援助により設立されたもので、Gopher、WWW、Z39.50⁶ や whois++などのネットワーク上の情報の発見、検索システムの普及をおこなっています。

Xgopher を例に Gopher を使ってみます。Gopher クライアントは、インストールする時点でどこをルートサーバーにするかを決めておきます。gopher.uiuc.edu などがルートサーバーとされていますが、Gopher の場合はリンクが張られているサーバーであればどこでもかまいません。近いサーバーや自組織のサーバーがあればそれを、そうでなければ gopher.nic.ad.jp を指定してお

図 3 Xgopher の例

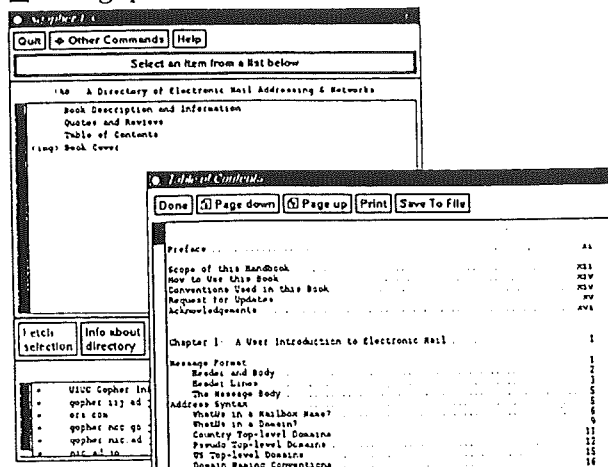
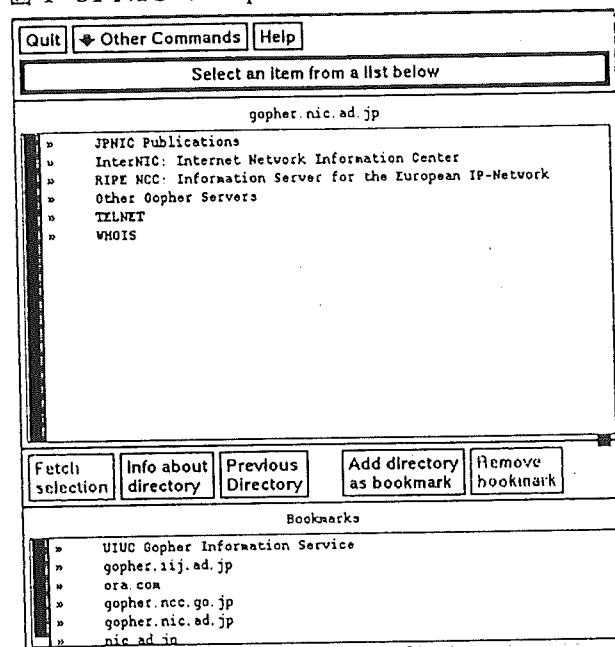


図 4 JPNIC の Gopher サーバー



くとよいでしょう。

インストール時の指定とは関係なく、

```
% xgopher gopher.nic.ad.jp
```

とすれば、最初に gopher.nic.ad.jp にアクセスにいきます (図 4)。Gopher の場合には、メニューを次の Gopher サーバーへのリンクにすることができます。この場合、さすがに JPNIC だけあって InterNIC と RIPE NCC がメニューのなかに入っています。

Gopher のなかからほかのプログラムを起動することもできます。この例が whois のエントリです。図 5では、Gopher から whois を起動して "shin@iij.ad.jp" を検索しています。

4 ftp.iij.ad.jp: ftp/pub/win3/util/gophbk11.zip

5 Microelectronics Center of North Carolina

6 WAIS で用いられているプロトコル。

図5 Gopher から whois を検索

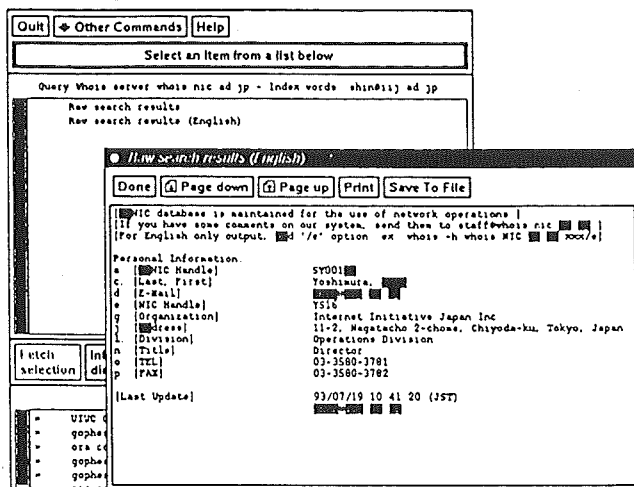
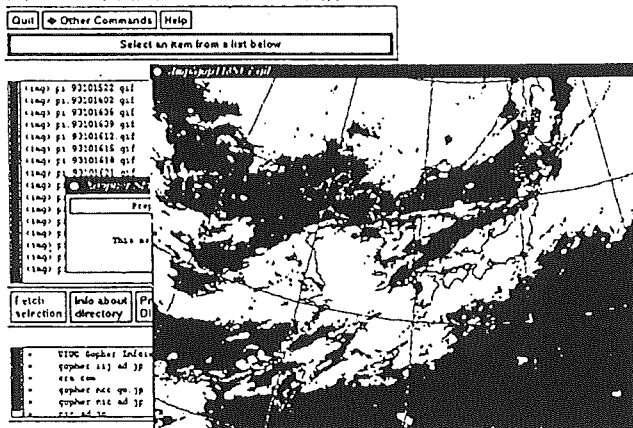


図6 気象衛星ひまわりからの画像



whois と同様に、anonymous FTP や telnet を起動して目的のリソースに到達することができます。

Gopher は文字情報だけでなく、イメージや音を扱うこともできます。文字情報にくらべると、イメージはデータ量が大きくなります。これが現在トラフィックを増加させている一因ともなっています。しかし、イメージや、音声はたいへん分かりやすく、魅力的です。消極的にならず、高速なネットワークが実際に必要とされていることを認識し、ネットワークを高速化する努力を皆でしましょう。図6は JWA (日本気象協会) が実験的に提供しているデータを、国立癌センターの Gopher サーバーで提供しているものです。台風が近づくと、アクセスが急増してネットワークが完全に潰れてしまうそうです。

WAIS

WAIS は、1993 年 5~6 月号の「UNIX Communication Notes」で紹介されていますから、今回は新しい

情報だけを紹介しておきます。

WAIS の話題としては、コマーシャル・プロダクトとして、WAIS サーバーと WAIS ワークステーションが発売されたことでしょう。WAIS はネットワーク・ホスティングとして位置づけられています。フリーのバージョンでも同様の特徴がありましたが、とくに優れた検索エンジンをもっています。これを商品としてさらにブラッシュアップして提供しようというものです。

プロトコルとしては、Z39.50 を採用していますから、従来のサーバーを利用することもできます⁷。

コマーシャル・バージョンのサーバーでは以下のような機能をもっています。

- さまざまなフォーマットに対する自動的なインデックス作成
 - UNIX mail、NetNews、rmail など
 - Microsoft Word、PowerPoint、Quicktime などのサポート
 - スプレッドシート、データベースとの変換
 - その他のドキュメントに対しても簡便な対応
- 1 つのソースに複数の入力データ・フォーマットをサポート
- ユーザー定義を含み、さまざまなソースをサポート
- 英語の語幹規則をサポート
- インデックスのインクリメンタル・アップデート
- インデックスのサイズの縮小 (ソーステキストの 1/3)
- 1 MB から 5 GB までのソース
- IP アドレスによりアクセス制限
- サーバーのロギングとサマリーのレポート機能

サーバーは、UNIX で動作します。シングル CPU で 15,000 ドルで、ソースコードは 40,000 ドルです。現在、SunOS 4.1.x、Solaris 2.1、NeXT で動作し、DEC Alpha、IBM RS/6000、HP への移植が予定されています。UNIX のほか、Macintosh、DOS、MS-Windows、NeXT など 12 種類のクライアントがあります。

これに対して、フリーのバージョンは現在 freeWAIS-0.2 となっています。

⁷ INET '93 ではプロトコルに手を入れることを考えているという話題も出ていました。



WWW と xmosaic

WWW (World Wide Web) は Gopher によく似たシステムですが、Gopher よりも互いのデータに柔軟な関係をもたせることができる、いわゆるハイパーテキストとなっています。

WWW のクライアントとしては、tkWWW や、ViolaWWW があります。今回は NCSA Mosaic を紹介します。

Mosaic は、X window、MS-Windows、Macintosh で動作します。Windows と Macintosh 版はまだ正式リリースではありません。Windows は最初の Gopher Book と同じように、Winsock 上で動作します。今回は日本語に対応したバージョンを作っている人がいるので、それも使わせていただきました。

Mosaic は、WWW のクライアントであると同時に Gopher にもアクセスできます。WAIS に関しては NCSA のゲートウェイを利用することにより使えます。

WWW によって実現されているハイパーテキストのイメージは、文章の説明より実物を見るのが早道です。カラーでお見せできないのが残念ですが、順に紹介していきます。

WWW のサーバーも最初からどこにあるかが分かっていたら、いきなりアクセスにいきます。これには、xmosaic の File のメニューのなかから Open を選び、<http://www-hostname> とするとアクセスできます。どこにあるか分からないときには、Documents のメニューから Other Web Documents のサブメニューにいき、Web Server Directory を選ぶと、図 7 の表示になります。下線付きのワードは次の画面に進むことができるアイテムです。

このなかから、本誌で連載している BSDI のサーバーを覗いてみました。図 8 は、BSDI のホームページです。ここから、最初の BSDI and BSD/386 Information を選択したのが図 9 です。この画面で、Rob Kolstad の顔写真とともに、彼の音声でのメッセージが入っています。音声を取り扱うことのできるワークステーションで動作させた場合、左の「スピーカーらしきアイコン」をクリックすると音声データが転送されて再生されます。

次に、BSD/386 Hypertext Man Pages を選んでみ

図 7 Web Server Directory

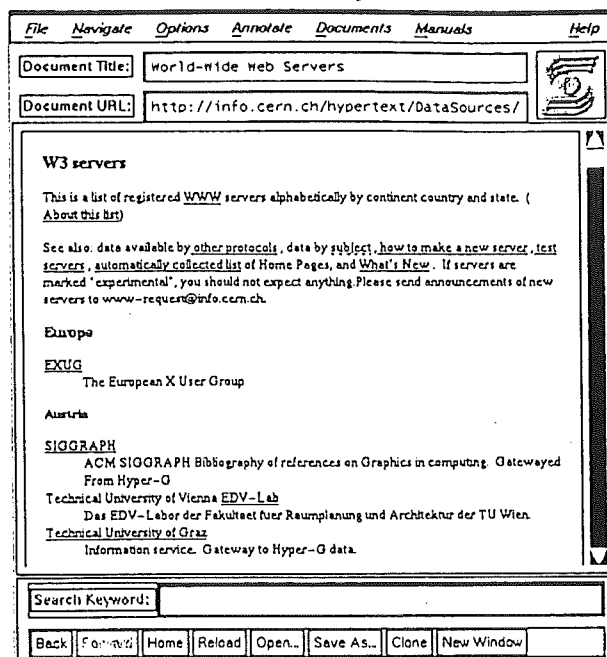
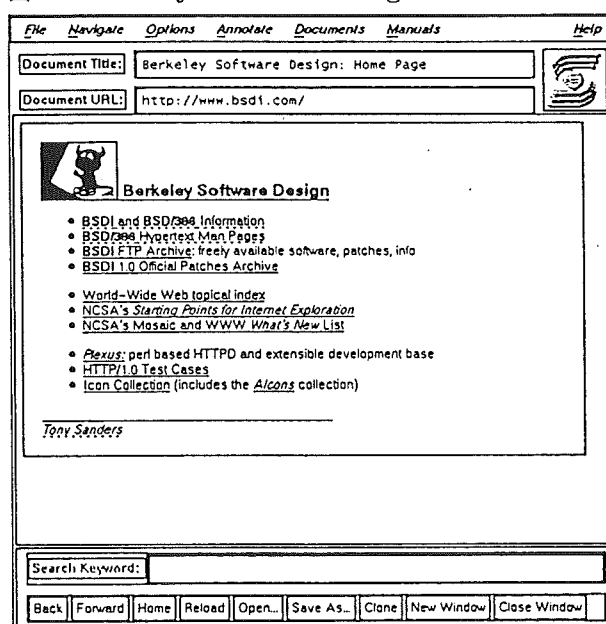


図 8 Berkeley Software Design のホームページ



ました (図 10)。ここで、下段の Search Keyword の入力をする、UNIX の apropos コマンドと同じ動作で、該当するコマンド一覧が現れます (図 11)。そして、下線のついているところをクリックすると、いわゆる man ページが現れます。

次にトライしたのが、O'reilly and Associates のサーバーです。ここには『Whole Internet Users' Guide and Catalogue』という本の巻末にある「Cat-

図 9 BSDI and BSD/386 Information

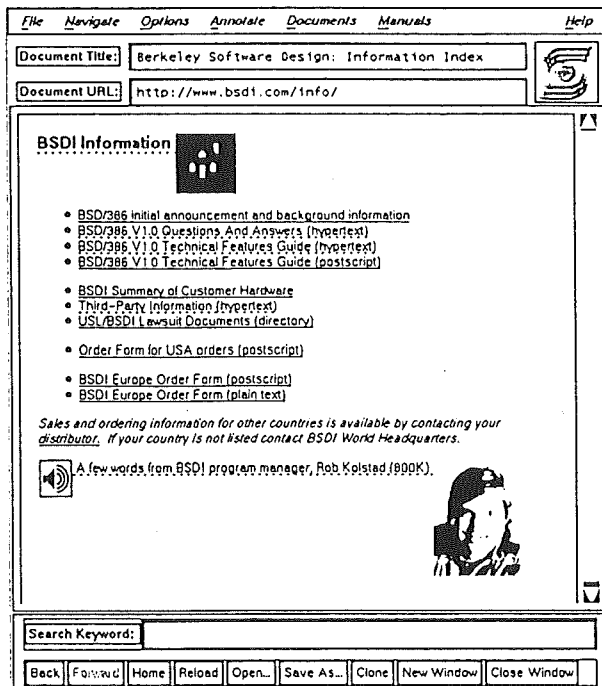
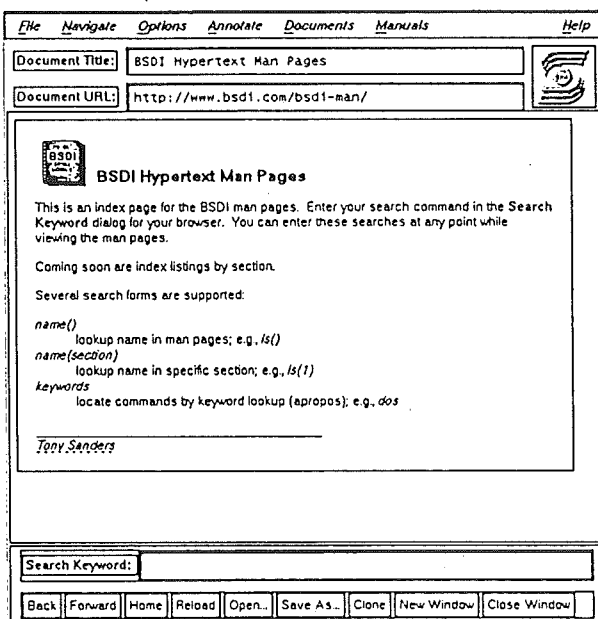


図 10 BSD/386 Hypertext Man Pages



atalogue」の部分が高ハイパーテキスト化されて置いてあります。

最初の図 12は、Science の Table of Contents ページです。ここから Astronomy を選ぶと、図 13になります。このような手順で順次サービスを選んでいくことができます。最後のはファイルを見ることになったり、telnet が起動されたり、Gopher が起動されたりして、ほかのインターフェイスを用いて情報にアクセスできるようになるこ

図 11 ping を検索してみたところ

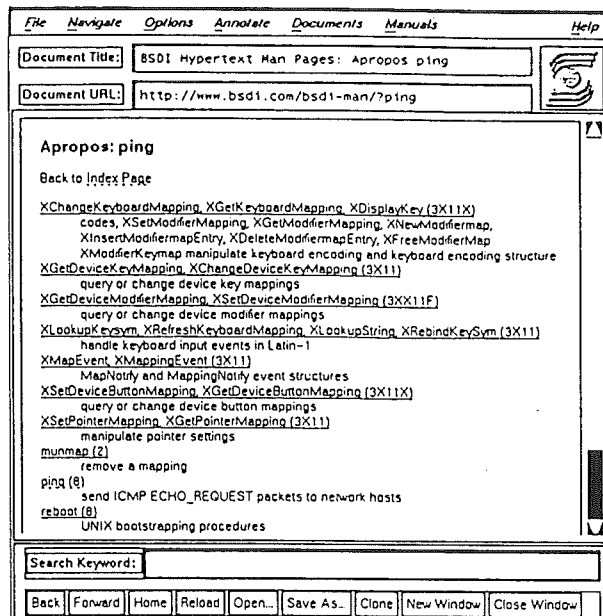
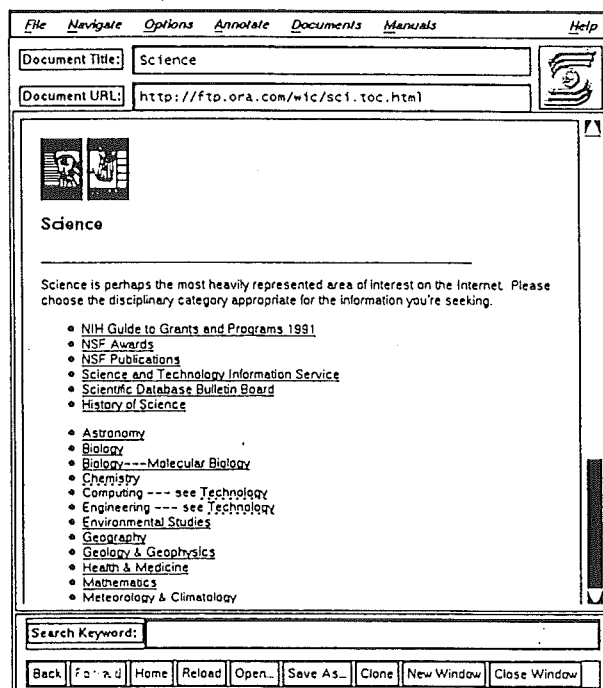


図 12 Whole Internet Online Catalogue (Science)



ともあります。

Gopher もそうですが、この WWW も知らず知らずのうちに世界中のネットワークを駆けめぐることになります。図 14はまた別の Table of Contents で、ここではレクリエーションに関するエントリが並んでいます。

図 15は、この Catalogue のなかから飛び出て、オーストラリアの WWW サーバーにアクセスしている様子で

図 13 Astronomy を選んだところ、次に選択するとそれぞれのサービスにアクセスしていく

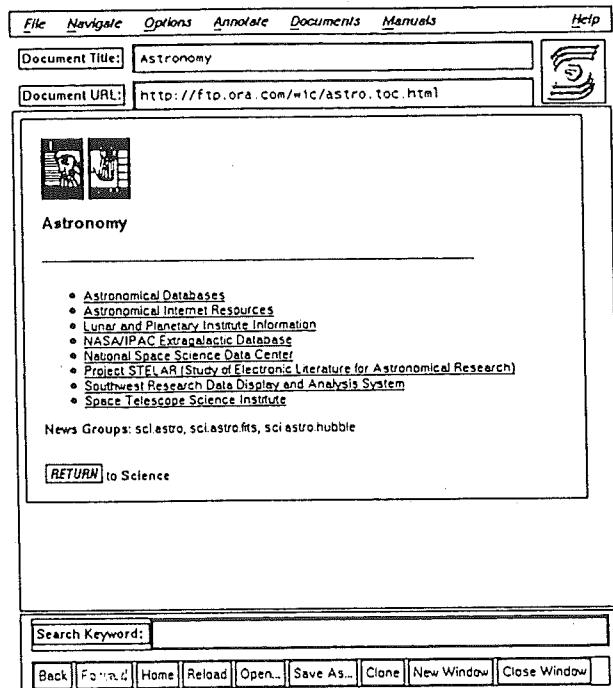
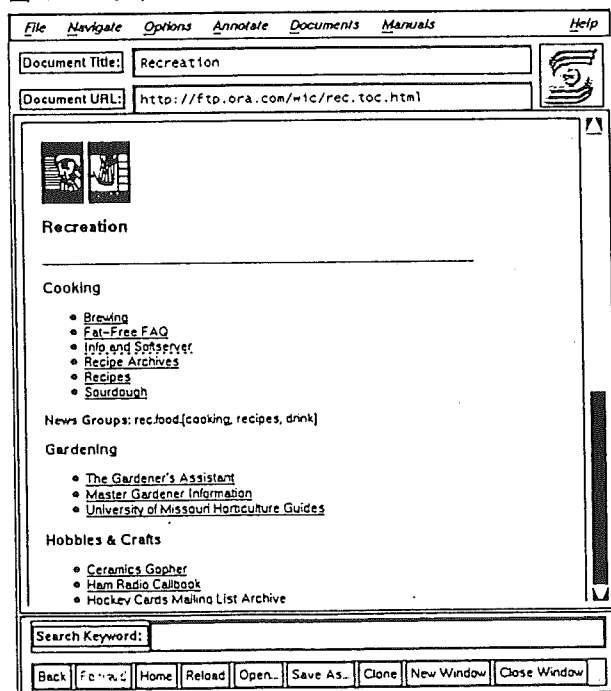


図 14 レクリエーションの Table of Contents



す。見た目の雰囲気あまり変わらないので、本当に世界のどこにいるか分からなくなってしまいます。

最後は日本です。日本語を含むものを試してみました。MH-6.8 の日本語化でもお馴染みの高田さんが管理している NTT のサーバーです。日本語の xmosaic に関して

図 15 Molecular biology を選んで、オーストラリアのサービスを検索にいった様子

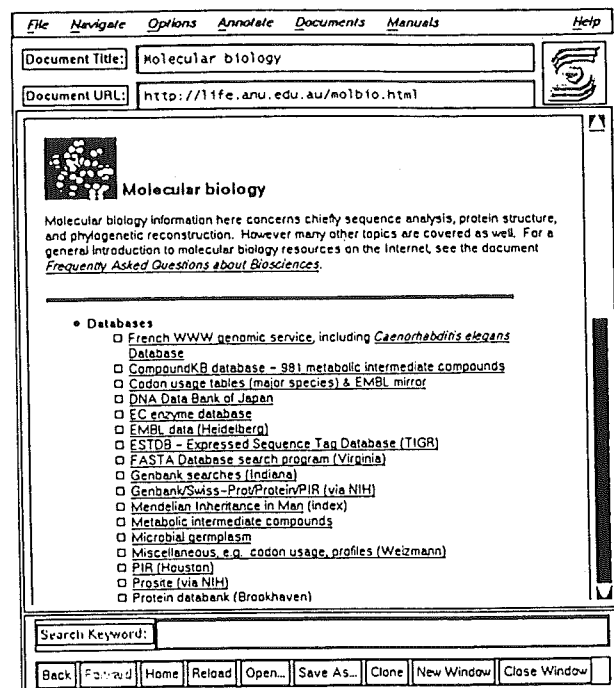
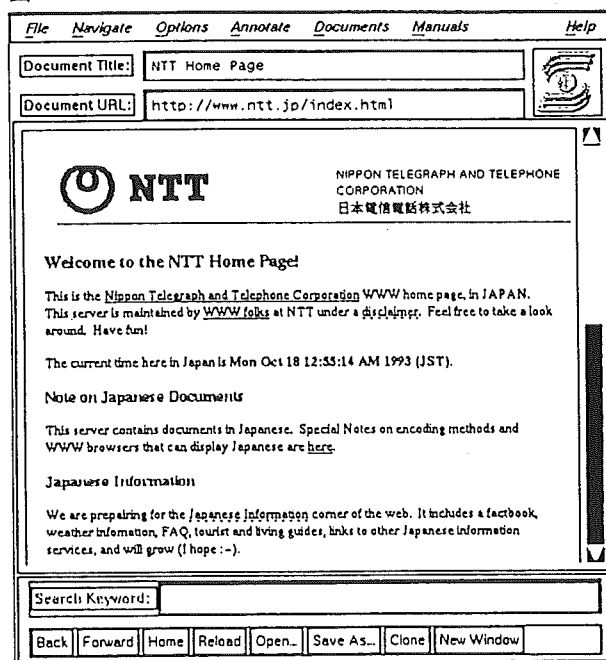


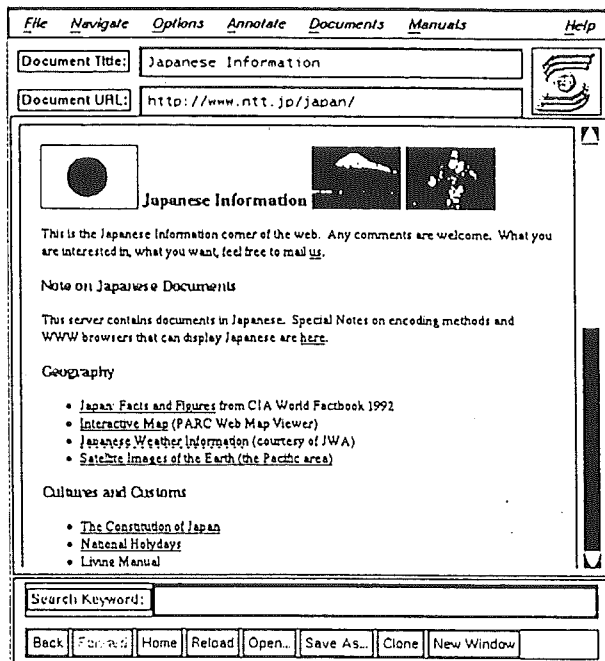
図 16 NTT Web server ホームページ



は、まだ日本語が通るだけだということですが、十分実用的です。

図 16 がホームページです。ここから Japanese Information を選ぶと図 17 に変わります。さらに Travellers' Japanese という項目を選んでみました (図 18)。ここには、簡単な日本語の言い回しが入っています。Shopping

図 17 Japanese Information



へと進んでいくと、買物のときに必要になる日本語の会話が紹介されています(図 19)。それぞれの日本語の部分には音声が入っています。

おわりに

今回は、デモンストレーション的に Gopher と WWW を中心に紹介しました。これを今後さらに詳しく紹介していこうと思って始めたのですが、じつはけっこうめづります。というのは、とくに WWW は簡単にカラーイメージを貼り込むために、データの転送に時間がかかるのです。64Kbps ではかなりこれは厳しい気がしました。私のオフィスは比較的回線の太いところなのですが、深夜自宅から INS64 で IP 接続をして原稿を書いています。この Gopher と WWW とをバリバリ使うのはとても大変でした。また日曜の深夜という国際回線が一番空いているときを見計らった作業でしたが、海外にアクセスするときには十分にひと息つけるくらいの時間がかかっています。

これをひろく紹介して、皆が使うようになったら回線が本当に潰れてしまいます。かといってこんなに楽しいアプリケーションをこっそり使うのは気が引けます。

しかし、冒頭に述べたような問題をできるだけ多くの人が知りかつ考えることが、現在インターネットによって実現され始めている知的空間を発展させていく力になるに違

図 18 Travellers' Japanese

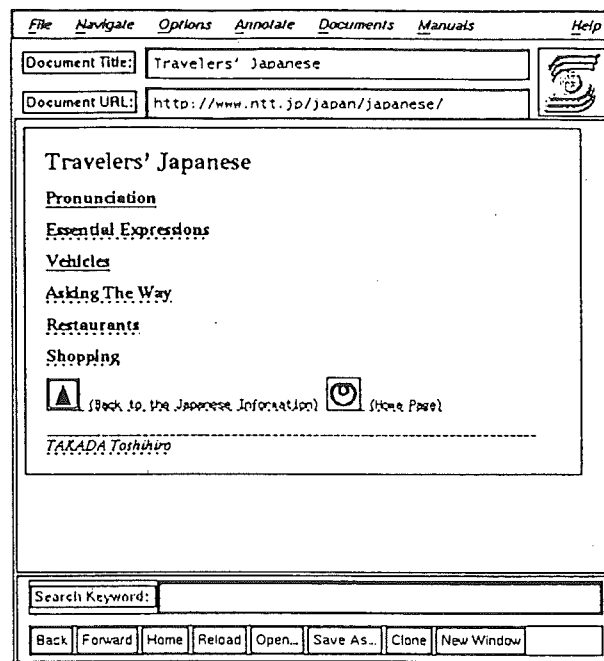
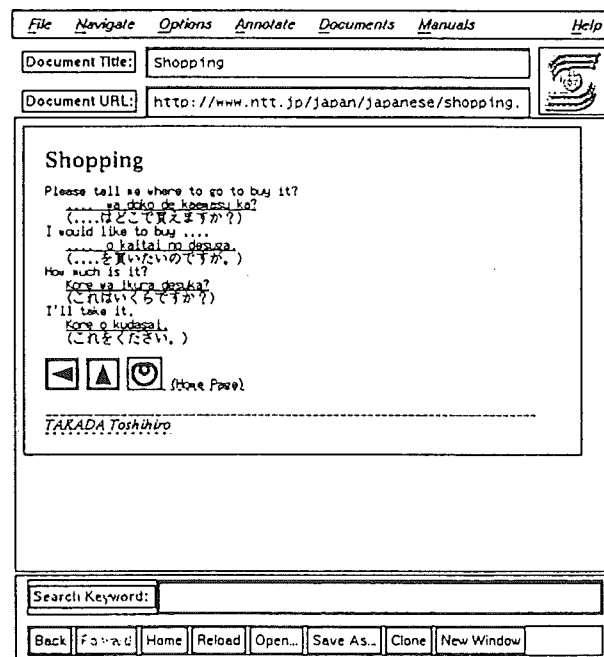


図 19 買物のときに必要な日本語



いないと思います。

次回から、それぞれの詳細について解説するつもりですが、ただ使うだけでなく仕組みを理解しつつ、このあたりの問題を頭の片隅において、利用者としてあるべき世界を考えてください。

..(よしむら・しん IIJ)

インターネットの
利用と仕組み

9

吉村 伸

Gopher

はじめに

日本は、世界一 ISDN (Integrated Services Digital Network) が普及している国です。では、ISDN はどのような使い方をされているのでしょうか？ 現在サービスされている ISDN は INS (Information Network System) 64 と INS1500 の 2 種類で、料金体系は電話と基本的に同じです。B チャンネル (64Kbps) 1 本あたりが電話 1 回線の計算になります。INS64 の場合、基本料金は電話 2 回線ぶんですが、DSU (回線終端装置) のレンタル料を加えると 5,000 円前後になります。

筆者は、INS64 を 2 回線使っています。用途は、64Kbps の IP 接続、19.2/9.6Kbps での非同期の TA を使うパソコン通信、FAX とモデムと電話です。しかし、INS ならではの機能を使っているものはほとんどないといってもいいでしょう。すべて TA (ターミナルアダプタ) という機器を使い、伝統的なインターフェイス (X.21、V.35、RS232C、電話) とのあいだの変換をおこない接続しているだけです。Integrate がなんでもつながるという意味なのは理解できますが、システムとして Integrate されている気がしません。もちろん、これは INS を上手に使う機器を用いていないからですが、そんな機器ほど恐ろしく高くここでは話になりません。

もっとも、INS を提供する側が使い方を十分に定義していませんから、ユーザーが素材として利用するのもやむをえないことでしょう。結果的には、電話 2 回線を 1 本の線で利用できる、X.25 を容易に使える、あるいは高速の伝送路とみるしかありません。料金面での利点を考えてみても、従来の電話で使えていたアプリケーションの移行では

なんのメリットもなく、高価な TA を購入しなくてはならないというデメリットだけということも多いでしょう。ただし、INS が使い物にならないわけではありません。コンビニエンス・ストアなどの POS システムや、通信量は少ないが短時間に伝送したいケース、トラフィックを明示的にコントロールできる使い方——たとえば、フロッピー・ディスクのコピーなどではある程度使えるようです。

筆者も 64Kbps で接続できるので ISDN を利用しています。しかし、ある時間を超過すると専用線の料金より高くなってしまいます。筆者の場合は、そろそろ専用線に切り替えたほうがよさそうです。

INS は、高価な設備投資を回収できるだけ付加価値の高い業務で、しかも通信量が少なく、トラフィックを予測しやすい専用のアプリケーションにおいてメリットがあります。

INS をインターネットとの関連で捉えると、専用線が安かったらまったく出番がないといわざるをえません。換言すれば、専用線が高いからこそかろうじて使い途があるだけです。INS がもつ交換網としての特性も、インターネット自身が接続できれば世界中のどこにでもつながるのでメリットにはなりません。

インフラストラクチャとして ISDN を考えたとき、各所へつながる電話という巨大な交換網としての利点もっています。しかし電話網と同じ技術、考え方で構築されているため電話の欠点もそのまま受け継いでいます。

インターネットは、もちろんそれ自身にどこへでもつながるという特性があります。しかし、その構築背景、利用法については電話網と大きな違いがあります。インターネットは、分散したサービスを継続的に利用する仕組みもっています。もちろんコネクティビティは、物理アド



レスによって構築されていますが、物理アドレスはトポロジーに依存します。これは電話でも同じです。

これに対して電話網では、電話番号は物理アドレスにすぎません。論理的な名前は電話帳にしかなく、網には論理的な名前とアドレスを自動的にマッピングする機能もあります。このため、電話番号が変わった場合、ユーザーはオフラインで電話番号を調べるしかなく、サービスを継続的に利用できなくなります。つまり、分散したサービスを管理する機能に欠けているのです。たとえば、今回 IIJ の UUCP サービスの電話番号が変更されましたが、これを教えてくれる仕組みは電話というネットワーク自体には存在しません。

インターネット資源のマッピング

ネットワークの目的の1つに、他のコンピュータ資源の利用が挙げられます。ディスクスペース、高速な CPU、データベース、プログラムあるいはプリンタの利用などがそうです。資源の利用だけでなく、離れたコンピュータのアプリケーションをつなぐことで、個々に独立しているメールボックス・システムを接続し、ほかのコンピュータ・システムのユーザーのメールボックスに書き込みをおこなったり、電子掲示板の記事を相互に乗り入れることもできます。

先月号の「UNIX Communication Notes」で紹介されたように、離れたコンピュータに接続されている FAX モデムを使うことも考えられます。

インターネットの利用は、独立したコンピュータを DNS (Domain Name System) によってマッピングすることにより、世界中のコンピュータ資源を活用し、情報を交換し、共有することにあります。DNS によって管理される名前は、物理的な位置や IP アドレスとは独立して、論理的な名前が与えられ、論理的な階層構造を構築しています。

利用者が必要とするのは名前だけであり、物理アドレスをいちいち知る必要はなく、アドレスの検索は IP ネットワークの基本機能として提供されているものを利用します。これにより、巨大なネットワークの急速な膨張と変化に対して振り回されることなく、ユーザーはサービスを継続的に利用できます。

じつは、IIJ の UUCP サービスの電話番号が変わるすこし前に IP アドレスも変更したのですが、このことに気づいた人、影響を受けた人は少なかったはずです。

Gopher

ユーザーが分散する情報を扱うのにアドレスを意識する必要はありません。シンボリックなアドレス形式ならまだしも、物理アドレスを意識することは、好ましいユーザー・インターフェイスとはいえません。

今回紹介する Gopher は、インターネットの提供する資源のマッピングと管理を応用して、世界中の情報を渡り歩くことを可能にするシステムの 1 つです。Gopher ではさらにこの抽象化を進め、ユーザーが分散情報を利用するのを容易にしています。

前回、現在のインターネットで情報を検索し、共有するための代表的な専用アプリケーションとして、Gopher、WWW、WAIS を紹介しました。

これらの特徴は、利用者が現在アクセスしている情報のある場所と、そこまでの距離と経路とを意識しなくてすむ点にあります。

Anonymous FTP の場合、ユーザーは明示的にアクセスする場所を指定します。したがって、多少なりとも自分がアクセスする場所を意識します。ドメイン名によってマシンを指定しますから、どこの国のサーバーか、あるいは国内のどこのサーバーかは名前から分かります。

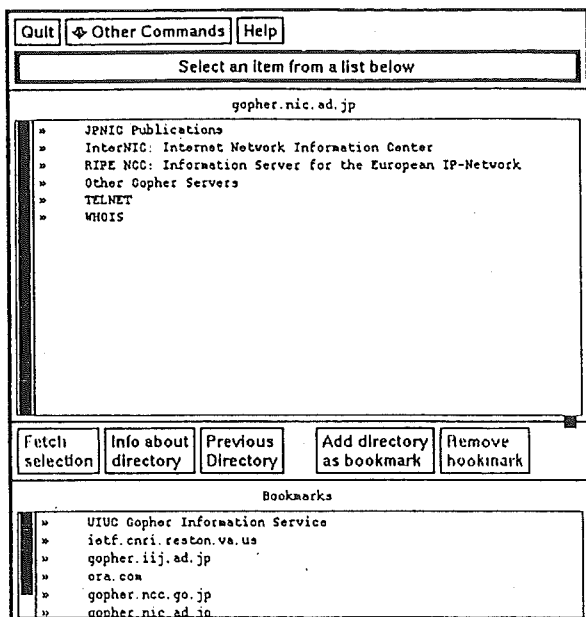
Gopher と WWW は、インターネット上の情報を使いやすいかたちでまとめるアプリケーションです。さらに、Gopher は、インターネット上のすべてのリソースをメニュー形式でブラウジングできるようにしたシステムです。

従来は、anonymous FTP を用いる方法ではすべてがファイルで置かれていて、転送してきたものに処理をして見ていました。Gopher は扱いやすく並べ替えたものといっても間違いではありません。

メニュー形式であることから、ユーザー側の端末機器に特別の要求は少なく、カーソル・アドレッシング可能なキャラクタ端末でも利用できます。

X をはじめとする GUI ではもっと使いやすくなります。ビットマップ形式のイメージ、PostScript のようにアプリケーションを必要とするものなどの情報にアクセス

図1 Gopher の例 (JPNIC)



できます。さらに、音声を扱えるデバイスが用意されていると、音声にもアクセスできます。

中身は同じものですが、メニューでファイル名ではなく名前を付けることができ、見やすく並べているだけですが、とても使いやすくなります。

Gopher のもっとも大きな特徴は、メニューの項目がそのサーバー上にあるファイルだけではなく、アプリケーションを起動したり、他の Gopher サーバーを指示しているところにあります。これによって、インターネット上に分散するサーバーを渡り歩くことができ、自分がどこにアクセスしているのか分からなくなるほど便利な機能です。

Gopher は、ファイルシステムの階層構造を見やすくして、階層構造をもった組織——たとえば大学の学部や学科などで、簡単にデスクトップのワークステーションを使って最新のドキュメントを公開するために作られています。凝ったアイコンなどのビジュアルな要素はありませんが、サーバーを設置するのがわりと簡単です。

Gopherの機能

Gopher の機能を、X のクライアントである Xgopher を使って紹介していきます。

図1は、JPNIC の Gopher サーバーにアクセスした様子です。ここでは、メニューに以下の項目があります。

図2 Gopher サーバーへのアクセス

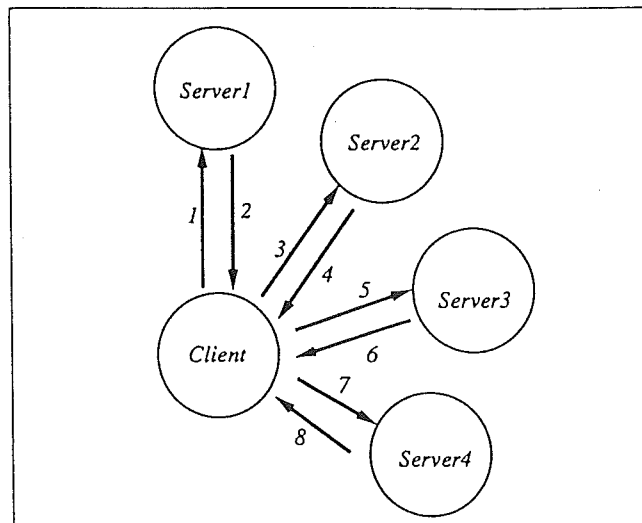
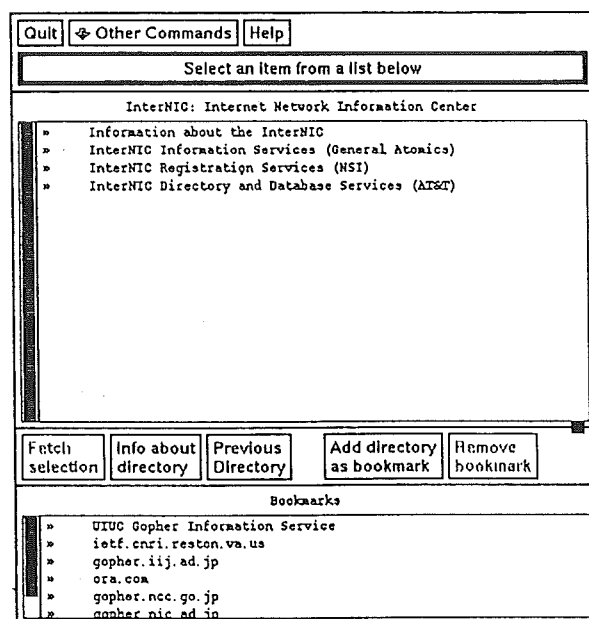


図3 JPNIC から、参照される InterNIC のサーバー



- JPNIC Publications
- InterNIC: Internet Network Information Center
- RIPE NCC: Information Server for the European IP-Network
- Other Gopher Servers
- TELNET
- WHOIS

最初の JPNIC Publications は JPNIC で提供しているドキュメント類で、JPNIC のサーバーに置かれているファイルを Gopher によって見られるようにしたもの

図4 メニューの各項目は、インターネット上のほかのサーバーへのポインタをもっている

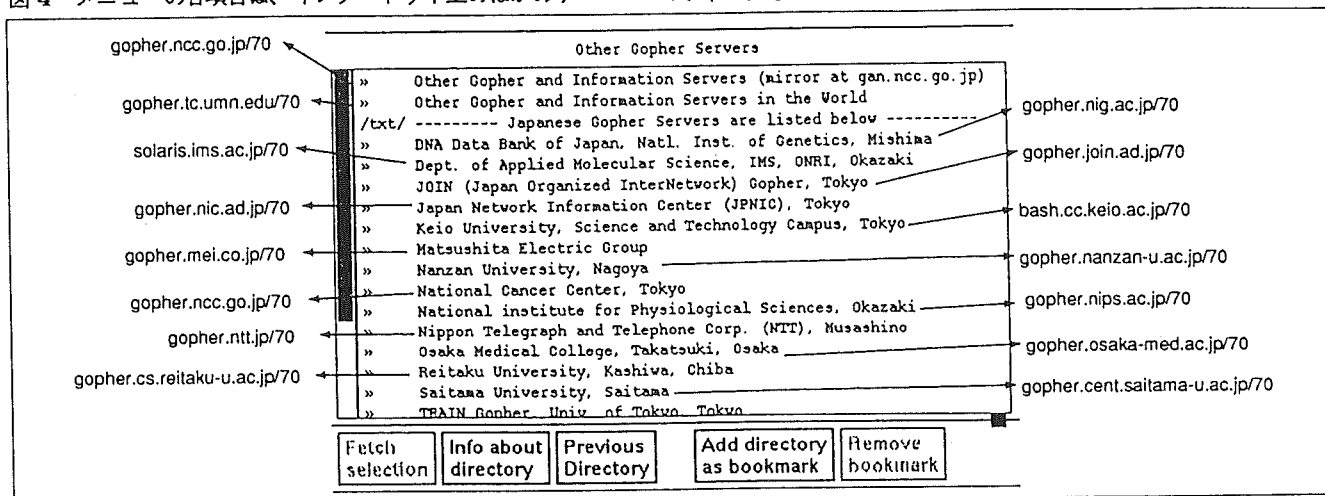


図5 Gopher サーバーのメニューの記述例

```

Numb=2
Type=1
Name=Other Gopher and Information Servers in the World
Host=gopher.tc.umn.edu
Port=70
Path=1/Other Gopher and Information Servers

```

です。

次の InterNIC、RIPE NCC は、それぞれの Gopher サーバーをアクセスにいきます。上にも述べたようにこれが Gopher の特徴で、サーバーが他の Gopher サーバーをアクセスすることを指示し、クライアントは指示されたサーバーにアクセスにいきます (図 2、3)。

WAIS でもそうですがインターネットでのサーバーへの問合せは、一般に、あるサーバーを経由して別のサーバーへアクセスするのではなく、サーバーを教えてもらってクライアントから直接アクセスします。したがって、利用者は全世界へのコネクティビティをもっていないと利用できません。

次の Other Gopher Servers はほかの Gopher サーバーの一覧です。これをアクセスにいくと、さらにメニューが出て個々の項目がほかの Gopher サーバーを参照するようになっています。それぞれのメニューが世界中のサーバーへのポインタをもっています (図 4)。個々のメニューは、アクセスすべきサーバー名、ポート番号 (アクセス先が Gopher であれば、通常 70)、アクセスした先のどのメニューの中身を表示するかを指示しています。上から 2 つ目の項目は、サーバー側の記述では図 5 のような簡単な記法で示されています。

表1 Gopher で扱われるテキストフォーマット例

拡張子	テキストタイプ	LOCALE
.txt.spanish	Text/plain	Es.ES
.txt.german	Text/plain	De.DE
.txt.french	Text/plain	Fr.FR
.ps	Text/postscript	
.tex	Text/tex	
.dvi	Text/dvi	
.troff	Text/troff	
.unicode	Text/unicode	
.sjis	Text/sjis	
.jis	Text/jis	
.euc	Text/jis	
.big5	Text/big-5	
.rtf	Text/rtf	
.word5	Text/MSWord5	
.word4	Text/MSWord4	
.mw	Text/MacWrite	
.wp	Text/WordPerfect51	

そのあとの、TELNET、WHOIS は telnet、whois によってデータをアクセスするエントリです。telnet で InterNIC のホストにアクセスしてデータベースの検索ができます。whois では whois コマンドで JPNIC、InterNIC、RIPE NCC の whois サーバーを検索します。

このように Gopher のメニューとしてほかのアプリケーションを起動し、結果を表示できるというたいへん柔軟

表2 RFC1436 に記述されている Item タイプ

0	テキストファイル
1	メニューエントリ、ディレクトリ
2	CSO phone-book サーバー
3	エラー
4	Macintosh の BINHEX 形式
5	DOS のバイナリ形式
6	uuencode されたファイル
7	インデックス・サーチ・サーバー
8	通常の telnet セッション
9	バイナリファイル
+	二重化されたサーバー
T	tn3270 セッション
g	GIF フォーマットのイメージファイル
I	イメージファイル、クライアントによっては扱えないこともある

表3 実験的に用いられているタイプ

s	Sun の audio (.au 形式) フォーマット
P	PostScript 形式 (クライアント側で、明示的に PostScript を表示するプログラムを実行する)
j	漢字テキスト (他のテキストとは区別して扱う)
M	MIME タイプ。MIME のメッセージ形式

な情報アクセス環境を提供します。

Gopherで扱うデータ形式

GopherのプロトコルはRFC1436に記述されています¹。

Item タイプ

Item タイプとして定義されているキャラクタを表2に示します。これによって、クライアントはどのように表示をおこなうかを決定します。Xgopher 1.3 のドキュメントには、これ以外にも実験的に使われているものとして表3のタイプが挙げられています。

テキスト

テキストは、LOCALE によって各国の言語を指示する仕組みをもっています。しかし、現状コード系に関しては他のフォーマット化されたテキストと同じように扱われています (表1)。これはプロトコル上の規定で、実際の取扱いはクライアント側の問題となり、どのような場合でも扱えることが保証されているわけではありません。

1 The Internet Gopher Protocol (a distributed document search and retrieval protocol). F. Anklesaria, M. McCall, P. Lindner, D. Johnson, D. Torrey & B. Albert. March 1993.

これらのフォーマットには、タイプ0 (テキストファイル) または9 (バイナリファイル) が割り当てられており、クライアント側でタイプによって起動するアプリケーションを切り替えます。

国内の Gopher サーバーには、タイプ0 (テキストファイル) として、日本語のテキストが JIS、EUC、SJIS のさまざまなコードで保管されています。この場合、たとえば jless-177 を使えば漢字コードを自動判別し、その端末に適した表示をするので簡単に見ることができます。Xgopher の場合には図6のリソースを設定します。

このように設定しておくと、X 自体が対応していなくても、日本語の表示ができます (図7)。

PostScript やほかの形式が、タイプ0 や9 で混在して提供されているのを表示できるようにするのは多少厄介です。しかし、表示できるプログラムが手許にあれば、ファイルの extension を見てプログラムを起動する Perl スクリプトなどを使って実現できます。

これは、Xgopher のドキュメントにサンプルが付属しています。たとえば .ps の場合、ghostview を起動するには図8のような Perl スクリプトを使い、これを Xgopher.type0.execCommand に指定します。これはテキストだけでなくほかの形式の場合にも利用できます。タイプ9 の場合にも同様の設定をおこなうとよいでしょう。

イメージデータ

イメージデータの多くは、GIF、JPEG です。GIF はタイプg としてとくに区別されていますが、X では xv や xloadimage など、さまざまなイメージ・フォーマットが1つのプログラムで取り扱えますから、これらを起動するようにしておけばうまく表示できます (図9)。

リソースの設定は、以下のようにします。

```
! --- use xv to view an image file
Xgopher.typeI.prefix: /img/
Xgopher.typeI.description: image file
Xgopher.typeI.execCommand: xv %f
```

テキストサーチ

タイプ7で定義されているインデックス・サーチ・サーバーは、WAIS と類似したテキストによる検索機能を提供しています。



図6 jless-177での設定

```

Xgopher.type0.prefix: /txt/
Xgopher.type0.description: text file
Xgopher.type0.dataType: Ascii
Xgopher.type0.execCommand: kterm -title "%n" -e less %f

```

図7 xgopherでの日本語表示

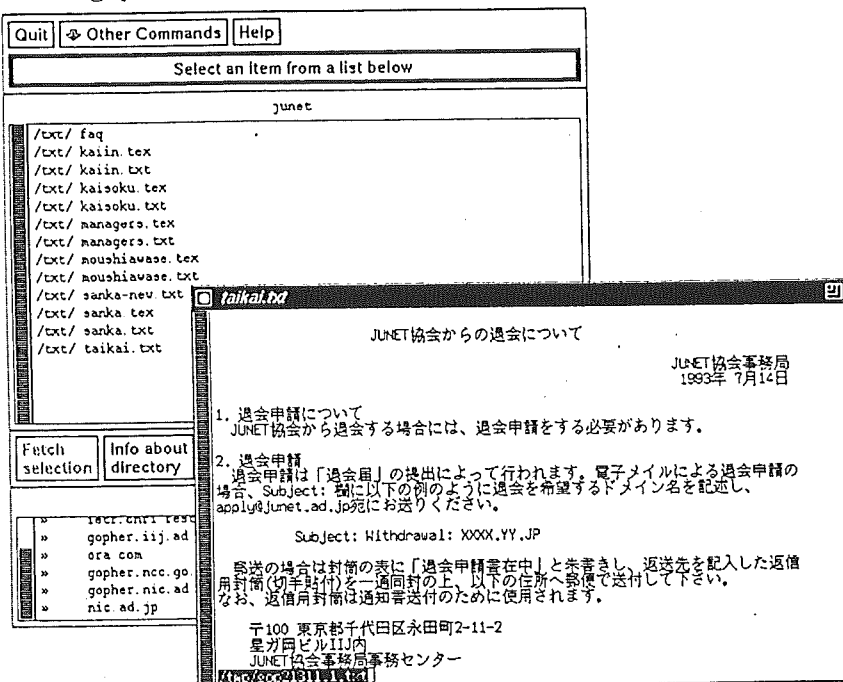


図8 Extensionによって表示プログラムを変えるPerlスクリプト例

```

#!/usr/local/bin/perl
if (! $ARGV[0]) {
    print "no filename specified!\n";
    exit 1;
}

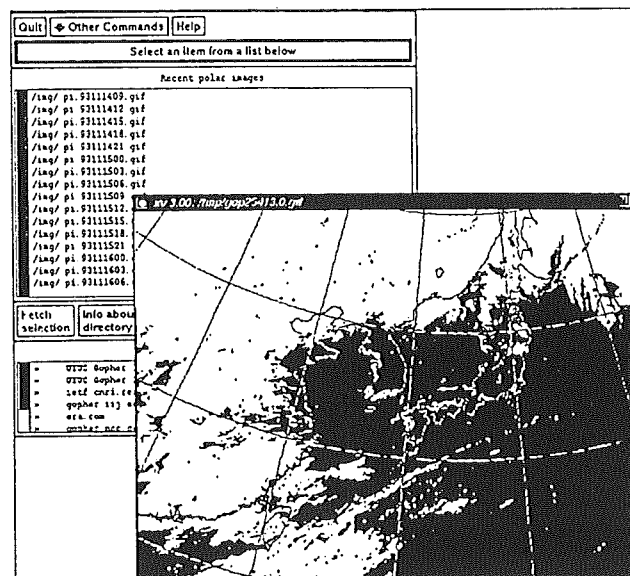
$dot = rindex($ARGV[0], ".");
if ($dot > 0) {
    $ext = substr($ARGV[0], ($dot + 1));
}
else {
    print "no extension parsed from $ARGV[0]\n";
    exit 2;
}

if ($ext eq "ps") {
    exec "ghostview $ARGV[0]";
}
else {
    exec "kterm -title $ARGV[0] -e less $ARGV[0]";
}

```

Gopherでは、Veronicaというプログラムがよく使わ

図9 xvを使ったイメージデータの表示例



れています。VeronicaはGopher itemのタイトルのインデックスを管理するPerlで書かれた簡単なプログラムです。これらのタイトルに対してキーワードで検索をおこないます。Gopherクライアントからの検索のリクエスト

図 10 Veronica によるインデックス・サーチ

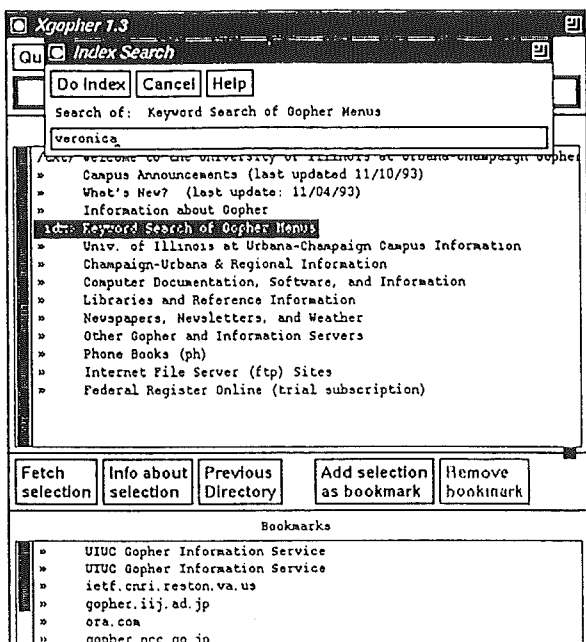


図 11 Veronica によるインデックス・サーチの結果

```
» Search titles in Gopherspace using Veronica
/txt/ About Veronica 11-17-92
/txt/ Veronica FAQ: Common Questions and Answers
```

を受け付け、結果を Gopher のメニューの形式で返します。したがって、ユーザーはこのメニューから次のアクセスができます。図 10は Veronica で検索するところで、図 11がその結果です。

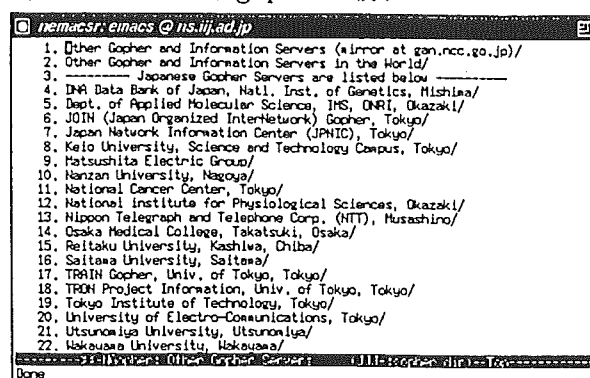
タイプ 7 で使われるのは、このほかには WAIS のサーバーを利用する場合や whois などをおこなう場合です。これらでは、実際には検索語を引数にしてシェル・スクリプトなどで whois や waissearch などのコマンドを起動し、結果をテキストとして表示します。

telnet でのアクセス

これはとても簡単な機構で、telnet を起動してネットワーク上のマシンにアクセスするだけのものです。テキストのクライアントでは直接 telnet コマンドを起動したり、Xgopher では xterm を起動し、telnet を実行します。

telnet の起動はタイプ 8 と T の 2 種類があり、後者は tn3270 を使います。3270 は、図書館の公開データベースなどでよく使われています。tn3270 は、IBM の 3270 ファミリーのフルスクリーン端末のエミュレーションをするコマンドです。SunOS などでは tn3270 コマンドは標準では付属していませんが、フリーで telnet と一緒に配

図 12 Emacs から gopher を使う



布されています。

他のクライアント

Gopher がよく利用されているのは、UNIX だけでなく、Macintosh や MS-DOS、MS-Windows などでも使えるクライアントのソフトウェアが多数あることにもよります。

UNIX

UNIX 用のクライアントとしては、まず Gopher² パッケージとしてサーバーと一緒に配布されているキャラクタベースのものがあります。このクライアントの場合にも、Xなどで使う際にはインストール時、あるいは gophercc ファイル (ホームディレクトリの .gophercc があればそれを使う) に、イメージ、dvi、音声などをどのように表示、印刷するかを設定しておきます。

きれいに日本語を表示するには、やはり日本語化された less などを使うのがよく、次のように指定します。

```
map: Text,less %s,jenscript %s
map: Text/plain,less %s,jenscript %s
```

1 番目がタイプ、フォーマット、2 番目が表示用プログラム、3 番目が印刷用プログラムです。イメージなどの取扱いも同様です。

Emacs ユーザーには gopher.el があり、Emacs から Gopher を利用できます。Nemacs、Mule を使うとほとんど日本語の心配をしなくてすむので快適です³。

2 現在 2.010。

3 ただし、Gopher は、Emacs が DNS を引くインストールにしてあることが必要です。



MS-Windows、Macintosh など

MS-Windows 用のクライアントは、前回紹介した Gopherbook と HGOPHER があります。これ以外にもいくつかあるようです。両者とも Windows の TCP/IP API の Winsock を使っています。HGOPHER には、PC-NFS のバージョンもあります。

Winsock は、MS-Windows 用の TCP/IP パッケージに付属しています。NetManage の Chameleon では動作が確認できています。

MS-DOS では、TCP/IP に対するインターフェイスが何種類もあります。きわめて複雑ですが、代表的な Packet Driver、PC/TCP、Lan Workplace 用のクライアントが存在しています。

Macintosh 用の実装は多数あります。NeXT ではもちろん UNIX 用のものも動きますが、NEXTSTEP を使ったものもあります。

驚くべきことに、PC、Macintosh のサーバーも存在しています。これらに関して確認することはできませんでしたが、インターネットの広がりを感じます。

VMS、OS/2 や IBM のメインフレームのインプリメンテーションも存在しています。

NCSA Mosaic

前回に紹介した NCSA Mosaic は、Gopher クライアントの機能も有しています。Mosaic は、X、Macintosh、MS-Windows 用があります。Mosaic を使うと、WWW、Gopher などのインフォメーション・ブラウザをほぼ全部使えるのでたいへん便利です。X 版の Mosaic は Motif が必要ですが、おもなアーキテクチャではバイナリが配布されています。

おわりに

Gopher では "Gopher Space" という言い方がよくされます。Space という感覚はとても的を射ていると思います。情報は距離、位置にまったく関係なく、あるスペースのなかに散在しているものということです。分散する情報をアクセスするのに、ユーザーはサーバーがどこにあるかは意識する必要がないものということを実感させてくれます。

Gopher は仕組みとしてはきわめて簡単なものです。しかし、機構が簡単なのでここまで多数のサーバーができたともいえます。いくつかの問題点、拡張をおこなうために Gopher+になっています。すでに多くのサーバー、クライアントは実際には Gopher+になっています (gopher2.010 は Gopher+です)。ユーザー側の使い勝手としては、Gopher+では MIME と互換性のある View タイプを指定できます。これにより、application/postscript や text/plain などの記述が有効に働いて適切な表示プログラムを起動できるようになり、見られるようになります。

Gopher を使ってみると、日本のインターネットは回線速度がまだ不十分で、現状ではイメージデータや音声データの頻繁な取扱いにはかなり無理がある、と感じました。

すこし前までは、anonymous FTP サーバーはできるだけ近くを使いましょう、国際回線はコストがかかるのでできるだけ国内で搜しましょう、ビジネスアワーを避けて使いましょう、というようにできるだけ行儀よくネットワークを使う努力をしてきました。これにくらべると、サーバーのある場所を意識しないことはずいぶん行儀が悪い使い方ともいえます。しかし、ユーザーニーズにあった使いやすいサービスは、ネットワークを作っている人にとってはわがままな行儀の悪いネットワークの使い方の方へ向かっていくかもしれません。

不幸にして、広域ネットワークが 2 地点間を結ぶ通信回線によって構築され、そのコストが大きく距離に依存しているという現実によって、インターネットのコストがゆがめられています。今後の新しい技術によって解消されていくことを信じたいと思います。

(よしむら・しん IIJ)

インターネットの
利用と仕組み

10

吉村 伸

WWWとインターネット上の情報探索

はじめに

早いもので、この連載も 10 回目を迎えました。この間に、日本のインターネットは大きく変わりました。インターネットは、1 年で倍増するといわるほど急速に成長しているので無理ありませんが、接続ドメイン数がついに 1,000 を超えました。

1993 年末にもいくつかの大きな変化がありました。1 つは、12 月 1 日から始まった地域ドメイン名のパイロット・プロジェクトです。従来、ドメイン名は組織だけに割り当てられてきましたが、地域ドメイン名の導入によってより広い範囲への割当てが可能になりました。これまで独立したドメイン名を取得できなかった個人経営の事務所、まったくの個人、家族なども、その割当てを受けることができるようになったのです。

地域ドメイン名は、yoshimura.shibuya.tokyo.jp というように、第 2 レベルに都道府県名または政令指定都市名を、第 3 レベルに市区町村名を、ローマ字表記¹でフルスペルで書くものです。これは、日本では米国のような 2 文字（または 3 文字程度）の一般的な略称コードがないためです。JIS コードや郵便番号のような数字を使わないのは、アドレスを間違えたときに他人に送られるよりも届かないほうがよいという判断からです。間違い電話は誰しも経験していることで、数字の羅列がとかく誤って入力されがちなのは理解できると思います。もちろん、今回はパイロット・プロジェクトですので期間中に各種の問題点を洗い出すことも目的の 1 つです。

地域ドメイン名の導入は個人だけではなく、地域密着型

の組織のためでもあり、小中高等学校や地方公共団体、図書館、各種厚生施設などが含まれます。高等学校がネットワークへの参加を検討し始めているようすし、都道府県の文化施設はこれまでにいくつかつながっていました。今後、このような動きが加速されていくでしょう。

インターネット接続をコマーシャル・サービスとして提供する会社も 3 社になりました。地域ネットワークの活動も活発です。このようなコネクティビティを提供するプロバイダの出現により、誰でもインターネットに接続することはできるようになりました。しかし、この連載で何回も指摘したように、回線費をはじめとするコストは個人が負担できるような額ではありません。

一方、昨年 12 月 8 日付で発表された NIFTY-Serve のインターネット・サービスは新しい時代の到来を予感させます。従来は、WIDE プロジェクトの実験として電子メールを交換していましたが、今度はコマーシャル・インターネット経由で IP 接続し、電子メールに続いてインターネット側から NIFTY-Serve へ Telnet でアクセスできるようになります（図 1）。これまでは電話をかけなければなりませんでした。今後はインターネットにつながっているワークステーションや PC からアクセスできるようになるのです。とりたてて画期的な試みとはいえませんが、実現されるときは極めて便利です。

将来は、NIFTY-Serve からインターネットへもアクセスできるようになるそうですから、個人ユーザーにもインターネットへのアクセスの道がひろく開かれるでしょう。さらに、個人向けアクセスサービスを提供するところがいっくつか出てきています。1994 年は、個人がインターネットに接続される年になるかもしれません。

1 原則的に使用大文字、リネンがない場合はヘボン式で綴ります。

図1 NIFTY-Serve への telnet

```
% telnet niftyserve.or.jp
Trying niftyserve.or.jp...
Connected to niftyserve.or.jp.
Escape character is '^]'.
Enter Connection-ID --->SVC/EUC
Enter User-ID --->SDI00403
Enter Password --->
```

```
ようこそNIFTY-Serveへ
Copyright (C) 1993
by NIFTY Corporation
All Rights Reserved
```

```
前回LOG OUT 93/12/05 22:27:02
ーメールが11通届いていますー
```

インテリジェント・ネットワーク

さて、インターネットというか、IP ネットワークには際立った特性があります。

インターネットがヘテロジニアスなネットワークによって成り立っている点です。そのなかには Ethernet や FDDI などのケーブリングがありますが、Ethernet ひとつをとっても、10Base5、10Base2、10Base-T の種類があります。さらに、リピータ、ブリッジによるネットワークの拡張もあります。レーザー光線や無線を使った接続もおこなわれているかもしれません。シリアル回線の接続をみても、構内の電話回線施設を使った SLIP/PPP からデジタル専用線によるものまで、その方式や質、速度は多種多様です。

しかしながら全体的に捉えると、インターネットとしての相互接続性を有した TCP/IP による通信が成立している、驚異的なネットワークとなっています。このような質の違うものを利用しながら、全体としてネットワークを構成できるからこそ今日の成功があるのです。

電話の問題点は、利用者がアクセスに数字の羅列を使うこと以外にもいくつかあります。その1つとして、電話の相互接続がきわめて均質な点が挙げられます。電話は、交換機という特定の機器間の相互接続で成り立っています。電話のもつこのような均質性の端局的な失敗例が、INS 網との相互接続です。

INS 網は、単独で完結するネットワークとして完成していればたいへん便利です。INS は、電話や FAX (G3、

G4)、デジタル通信などの属性をもち、同じ属性どうししか接続しないという性質があります。つまり、INS を正しく使うと、間違って FAX につながるようなことがないばかりか、同じ番号を使って電話と FAX を使い分けることもできます。しかし、既存の電話網で利用されている電話と FAX を相互接続したのが欠点です。

既存の電話網では電話と FAX が同じ電話回線に接続されており、INS に接続されている電話と FAX を識別できずに両方とも着信してしまいます。せっかくの Integrated Service であるにもかかわらず、けっきょく電話網と接続したために、利用者はダイヤルイン番号(付加料金が必要)を付けなければ電話と FAX を使い分けることができません。これでは、結果的に割高になり利用が増えないでしょう。電話と FAX がアプリケーションであるとすれば、最低のインプリメンテーションだといわざるをえません。プロトコルもへったくれもなく、針金の両端に無節操に端末をつないでいる“無手順”通信です。データを送るからあとは勝手に受け取ってくれ、としているわけです。ヘテロジニアスなネットワーク上にアプリケーションを構築できていません。

これが電話のような集中型ネットワークの欠点です。すべての機能が交換機に集中し、端末にはネットワークの制御機能をまったく期待していません。電話と FAX の例にしても、電話や FAX が「私は電話よ」「俺は FAX だ!」という機能があって、「あなたは電話ですか?」「おたくは FAX?」と“つながる前に”ネゴシエーション²すればいいと思いませんか?

INS にはこの欠陥を解消する機能があります。しかし、INS 網特有の接続機器自体がその機能をもち、本当の末端の端末どうしてネゴシエーションをしているわけではないので、電話と INS が相互接続するとうまくいかない部分が際立ってきます。

コンピュータ・ネットワークでは、ネットワーク自身と端末機器がインテリジェンスをもっていますから、

「もしもし、電子メールを送りたいんですけど」

「ごめんなさい、私は受け取れないの、〇〇に送ってね」

「〇〇さん、××さん宛の電子メールを送りますけど、い

² じつは、FAX にこの機能があります(旧い機種にはありません)。発信側の FAX からある信号を出し、これで留守番電話 FAX などの機能を実現しています。しかし、モデムが入ると事態は複雑になります。



いですか」

「はい、どうぞ」

というネゴシエーションをしてから通信を始めます。

TCP/IPの基本機能とDNS

このネットワークのネゴシエーションに相当する部分が、現在のインターネットではTCP/IPの技術におけるDNS (Domain Name System) と、TCP および UDP のマルチプレクシングです。

マルチプレクシングとは、TCP あるいは UDP でポート番号によってアプリケーションを識別する方法です。これは、INS の電話と FAX を識別するのと同様に、

「電子メール、送りたいんですけど」

「はい、どうぞ」

とか、

「FTP するけど、いい?」

「じゃあ、ftpd を呼びます」

という調子で、それぞれのアプリケーションがポート番号によって指定される仕組みです。

現在のインターネットを支える基本技術のなかで、TCP/IP のアプリケーションとして一番大きな役割をもつのは DNS でしょう。DNS も TCP、UDP のアプリケーションとして実現されていますが、それ以上の役割を担っています。

たとえば、

「もしもし、電子メールを送りたいんですけど」

「ごめんなさい、私は受け取れないの。〇〇に送ってね」

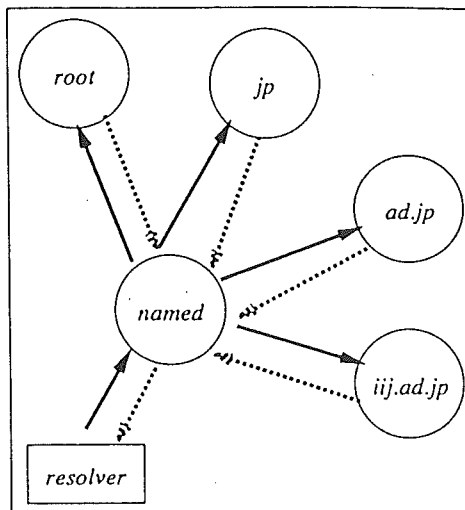
という部分は DNS がこなっています。

DNS

インターネットのリソースへアクセスする際、DNS は非常に大きな役割をはたしています。DNS の役割、仕組みの簡単なおさらいをしましょう。

DNS は、階層的ドメイン名と IP アドレスのマッピングのほかに電子メールに関する情報を管理しています。それ以外にもユーザーに関する情報などを管理することも規

図 2 DNS の再帰的問合せ



定されていますが、現在のインターネットでは一般に使われてはいません。

DNS は "." (ドット) で区切られた各階層を zone として扱い、上位 (右側) から順次下位 (左側) へ向かって検索していきます。

このとき、検索を必要とするプログラムは、resolver ライブラリによってあらかじめ設定されている最寄りのサーバーに問合せをおこないます。もし、そのサーバーが何も知らなかったときはルートサーバーに問い合わせます。ルートサーバーは問合せの上位の zone から順に判断して、知っているものを答えます。たとえば、ij.ad.jp の情報の検索で、もっとも手数のかかるケースでは、図 2 のように順に各 zone のサーバーを教えられ、それに従って最終的に ij.ad.jp の zone を管理しているサーバーから答を得ます。途中の答からは、次の下位の zone のサーバーのアドレスしか得られません。最終的な回答は、個々のデータを管理しているサーバーから直接得ます³。これは、インターネット上の分散リソース管理の基本的な考え方です。そのため、自分自身がインターネット全体への接続性がないと、最終的な回答を得るのに失敗します。

DNS では、名前に別名を与える機能があります。最近、多くの anonymous FTP サーバーは ftp.yyy.ac.jp という名前になっています。これが本当のホスト名であることは稀で、一般には別名になっています。

電子メールに関する情報も DNS で管理されている重要

³ DNS では検索結果をキャッシュしています。たまたまキャッシュしているデータがあれば、その答を返すこともあります。



なものの1つです。「メールを送りたいのですが?」「私は受け取れないので、〇〇に送ってください」というような機能は、MX (Mail eXchanger) によって実現されています。これは、電子メールを直接受け取れないマシンや、IP で接続されていない場合に別のマシンで受け取ってもらう機能です。

本誌 1993 年 12 月号の「UNIX Communication Notes」で紹介された TPC.INT も、DNS の名前空間の柔軟なマッピング機能を応用したものです。

この記事の説明では、FAX 番号 +1 415 123 4567 へ FAX を送るのに 7.6.5.4.3.2.1.5.1.4.1.TPC.INT というドメインを用います。このとき、DNS の設定は以下のようになっています。

```
$ORIGIN TPC.INT.  
*.5.1.4.1 IN MX 10 dbc.mtview.ca.us.
```

これは、+1 415 xxx yyyy という電話番号すべてにマッチします。すなわち 415 (カリフォルニアのサンフランシスコ湾北部周辺) の番号で始まる地域のローカルコールでつながるエリア全体という意味になります。

DNS によって管理される名前空間の応用は、このようにきわめて柔軟です。だからこそ、インターネット上に分散するリソースの活用とサービスの継続的な利用のために DNS が不可欠なのです。階層的概念を適用できれば、インターネット上のリソースからインターネット外のリソースまでをも関係づける可能性をもっています。

そして次にあげる URL への発展がこの延長であり、DNS はその要素の重要な 1 つになっています。

Uniform Resource Locators

ネットワークは机上の設計どおりに作れるものではありません。そもそも、動いている異なるサービスを接続してこそ意味があり、できるだけ整合性を保ちながらユーザーにとって使いやすい環境を構築しなければなりません。

ここにインターネット上の分散したリソースに対して、統一的な名前の付け方を決めてユーザー・インターフェイスを向上させる試みがあります。それが URL (Uniform Resource Locators) です。そして、この URL が今回の 1 つのキーポイントです。URL は、IETF の URL

ワーキンググループ⁴ で議論されているものです。現在、InternetDraft としてドキュメントが出ています (draft-ietf-uri-url-01.txt)。

現時点で対象として挙げられているものには、WWW で使われている HTTP (HyperText Transfer Protocol)、FTP (anonymous FTP)、NetNews、WAIS、Prospero、Gopher、Telnet、rlogin、tn3270、X.500、whois などがあります。

URL は、WWW から使われ始めたフォームを発展させたものです。リソースを特定する要素として、ホスト名 (ドメイン名)、ファイル名、アクセスに用いるプロトコル、アクセス時のコマンド列など最終的なドキュメント、ファイルに到達するまでのさまざまなオペレーションがあります。これらを取り込んで 1 つの名前空間を作ろうというものです。これは、

```
scheme:path[?search]
```

という形式が基本です。scheme はアクセス手段、すなわちプロトコルです。これには、HTTP、Gopher、Wais、FTP、Telnet、afs などがあります。path はアクセス手段によって異なりますが、リモートホストの名前が最初に来て、//remote.foo.bar.edu などのように “//” に続けてドメイン名を指定します。検索をおこなうときは、“?” に続けてキーワードを指定します。

このようにして、インターネット上のリソースに名前を付けるルールを規定しています。以下、各アクセス手段ごとにすこし細かく説明します。

File/FTP

たとえば、ftp.iiij.ad.jp で anonymous FTP 公開されている /pub/info/Index というファイルは、URL では次のように表されます。

```
ftp://ftp.iiij.ad.jp/pub/info/Index
```

また、次のようにすると “pub” ディレクトリを表します。通常、そのディレクトリ一覧が相当します。どのよう扱うかはアプリケーションに委ねられます。ただ表示するだけであったり、メニューとして加工したりします。

```
ftp://ftp.iiij.ad.jp/pub
```

⁴ ietf-url@merit.edu. 参加申込みは ietf-url-request@merit.edu.



Gopher

デフォルトのポート番号 70 で提供されている場合には次のようになります。

```
gopher://gopher.nic.ad.jp/
```

70 以外のポートで提供されているときは次のように、ドメイン名のあとに ":" (コロン) で区切って、ポート番号を示します。

```
gopher://gopher.foobar.edu:1234/
```

Gopher の場合、ここからさらに深くするのは少々複雑です。前回解説したタイプがこのあとに続き、さらにメニューが続きます。

```
gopher://gan.ncc.go.jp/11/CNET
```

NetNews

通常、NetNews は世界中に流れているのでサイト名は指定しません。ニュースグループ "fj.sources" は次のように表されます。ニュースサーバーは、環境変数 NNTPSERVER を参照するなどします。

```
news:fj.sources
```

HTTP

HTTP は、WWW の中核をなす HyperText Transfer Protocol です。サーバーとして、www.ntt.jp を示す場合は次のようになります。

```
http://www.ntt.jp
```

サーバー上のファイルは次のように表されます。

```
http://www.huie.hokudai.ac.jp/hokkaido/jp.html
```

Telnet

Telnet も、データベース・アクセスの手段としてまだまだ使われていますし、もちろんもっとも簡単なリモートアクセスの方法です。

```
telnet://rs.internic.net
```

これも同様に、デフォルトの Telnet のポート番号 23 以外では Gopher の場合と同じようにコロンで区切ってポート番号を続けます。

WAIS

WAIS には、インデックスによる検索機能があります。

```
waiss://sunsite.unc.edu:210/sunflash-1993
```

```
waiss://sunsite.unc.edu:210/sunflash-1993?DAT
```

上の行は sunsite.unc.edu のポート 210 で運用されている WAIS サーバーの sunflash-1993.src というソースを、2 行目はそのソースから DAT というキーワードで検索した結果を示しています。

WWW のクライアント・ソフトウェアは、この URL 表記によって各リソースへのアクセスができるようになっています。しかし、どの種類のリソースにアクセスできるかは、ソフトウェアによって異なります。

WWWとMosaic

WWW

WWW は、スイスにある CERN (Organisation (旧 Conseil) Européenne pour la Recherche Nucléaire) で最初に開発されました。

インターネット上の情報をハイパーテキストとして見られるようにしようという試みで、この中心となるのが HTTP です。HTTP は、HTML (HyperText Markup Language) で書かれたテキストを解釈し、ネットワークを通じてクライアント側へテキストやイメージなどを送出して、タグに指定されている URL を教えます。HTML の記法は、SGML (Standard Generalized Markup Language) をベースにしています。

Gopher では、次のリソースを示すタグはメニューですが、WWW では任意の文字列やイメージ (アイコン) を使うことができます。これがハイパーテキストといわれる所以です。Gopher と同じように、タグはローカルのリソースだけでなく、インターネット上のほかのリソースも指示しています。図 3 のように各タグの指すものは URL 記法で表され、HTTP だけでなく、Gopher やほかのものもすべて含むことができます。

WWW のクライアントとしては、テキストベースのもの、GNU Emacs 用のものなどあります。X ウィンドウ用のクライアントとしては、本誌の「Little Language」

図3 ハイパーテキストのタグがほかのリソースを指し示す

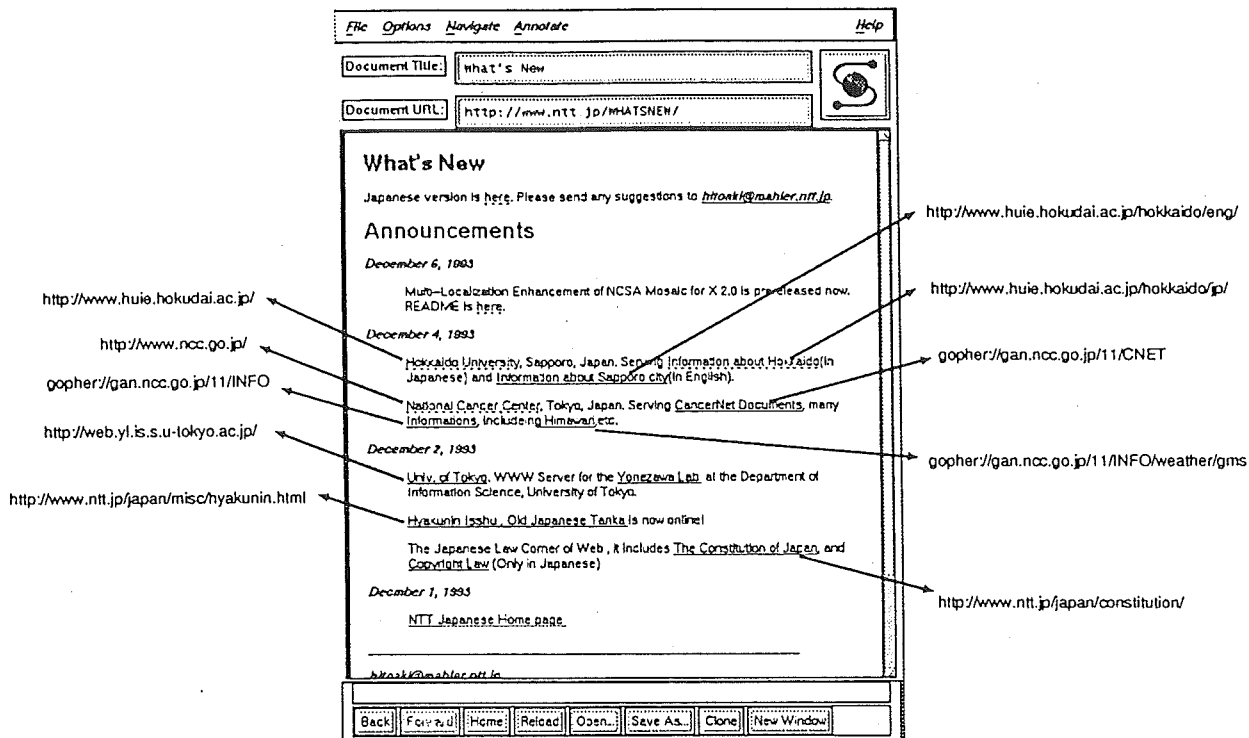
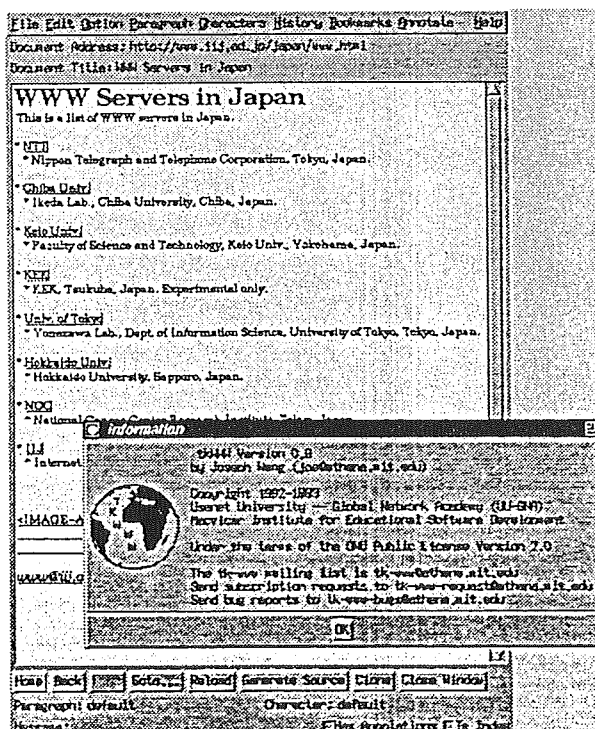


図4 tkWWW



で紹介された (1993 年 6 月、10~11 月号) Tcl/Tk で書かれた tkWWW (図4) などがあります⁵。

⁵ 現在バージョン 0.9 です。Mosaic には Motif が必要ですが、こちらはすべてフリーのツールで構築できます。しかし、機能は残念ながらまだ

Mosaic

しかし、いまや NCSA (National Center for Supercomputing Applications) で作られた Mosaic が事実上の標準となっています。ほかのクライアント・ソフトウェアも、Mosaic のオペレーションとの互換性を意識しています。

現在のところ、Mosaic は X ウィンドウ・システム、Macintosh、MS-Windows で利用できます。その使い方はきわめて多彩で、今回だけではとても説明しきれません。

X 用の Mosaic には Motif が必要ですが、主要なアーキテクチャのマシン用にはバイナリが配布されています (現在、バージョンは 2.1 です)。

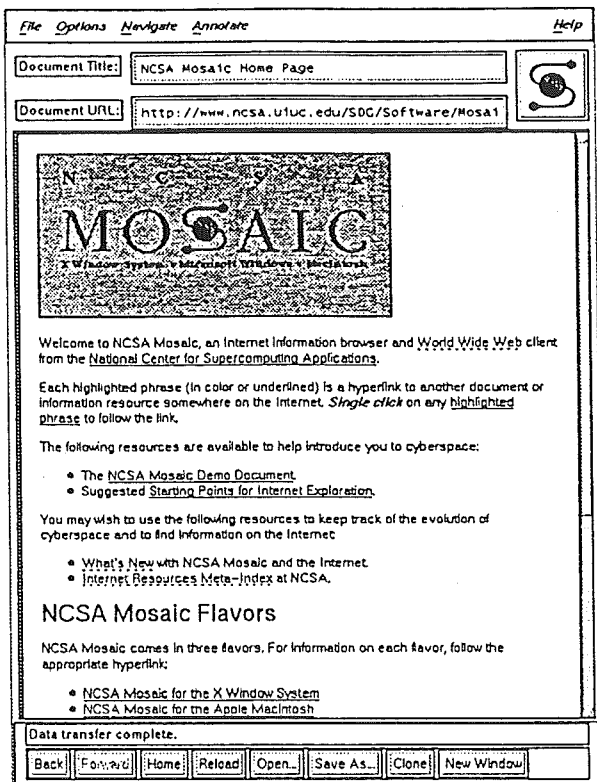
Mosaic では、イメージをインラインで表示したいへん美しい画面が作れます (図5)。

HTMLとMosaicによる表示

WWW の美しい画面設計は、HTML によって実現されています。まず、ごく簡単な例から紹介します。

省略です。

図 5 NCSA Mosaic の Home Page



```
<TITLE>The simplest HTML example</TITLE>

<H1>This is a level one heading</H1>

Welcome to the world of HTML.
This is one paragraph.<P>

And this is a second.<P>
```

これを、Mosaic で見ると図 6 のようになります。

ほかのドキュメントに対するリンクは、次のように記述します。

```
<a href="http://www.iiij.ad.jp/">IIJ</a>
<a href="gopher://gopher.nic.ad.jp/">JPNIC INFO</a>
```

この場合、“IIJ” “JPNIC INFO” という文字列の色が変わったり、下線が付いたりします。そこをマウスでクリックすると、前者では HTTP によって www.iiij.ad.jp へアクセスし、後者では Gopher プロトコルで [gopher.nic.ad.jp](gopher://gopher.nic.ad.jp/) へアクセスします。Gopher へアクセスした場合は Gopher Menu を生成し、メニュー形式のアクセスをします (図 7)。

href= の後ろには、URL 形式のリソースを指定しま

図 6 簡単な HTML の記述例

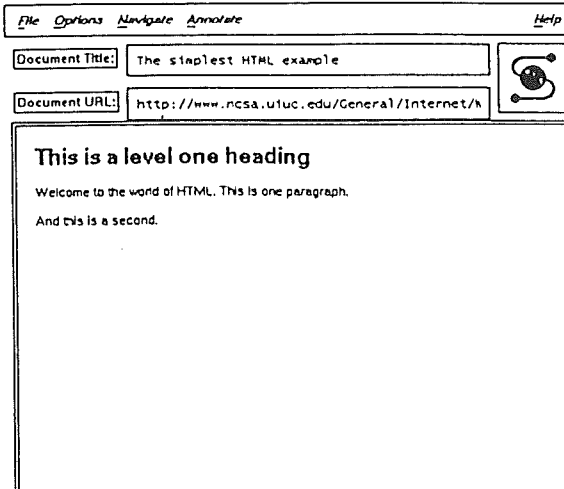
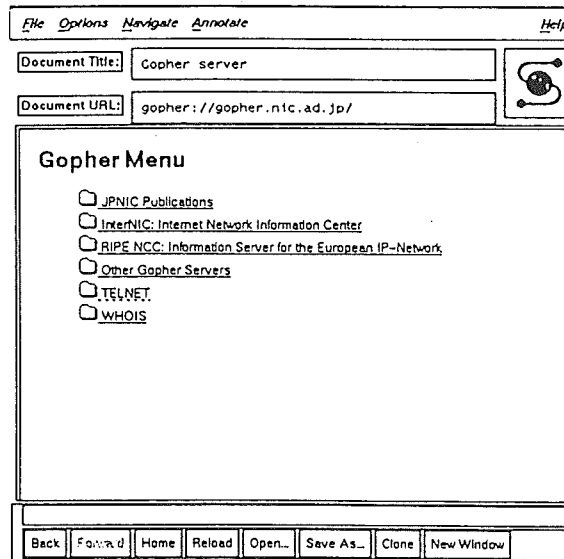


図 7 Mosaic による gopher server へのアクセス



す。すなわち WWW とは、HTTP (HTML) によってさまざまな URL により表されるインターネット上のリソースに対して、ハイパーテキスト・アクセスを可能にしたものであるといえます。

URL から指示されるリソースへのアクセスは、クライアントに使うソフトウェアによって異なります。Mosaic 2.0 の場合、WAIS に対してもゲートウェイを介さずに直接アクセスできるようになっています。

HTTP

html で書かれたファイルを解釈してネットワーク側からのリクエストに応えるのが、HTTP です。HTTP は現



在、Internet Draft (draft-ietf-iiir-http-00.txt,ps) として公開されています。

HTTP のサーバーは現在、NCSA httpd と、Plexus (Perl で書かれた HTTP サーバー) など複数の実装があります。

Mosaic と HTTP サーバーのあいだの通常のやりとりはきわめてシンプルです。Mosaic は、まずどこかの httpd に接続してセッションを開始します。このときに "GET /" というコマンドを送ります。すると、HTML で記述された Home Page が送られてきます。

Mosaic 側がサーバーから送られる html で書かれたファイルを解釈して、必要なインラインのイメージやタグ情報を得ます。ほとんどの解釈はクライアント側でおこないます。時刻など、一部のダイナミックな情報はサーバー側で処理されて送られます。

おわりに

WWW、Mosaic、Gopher の仕組みがきわめて単純であることには驚かされます。たとえば、HTTP はポート番号 80 ですから、Telnet を使ってサーバーに接続してみると実際に検証できます。

HTTP では、HTML で書かれたものがそのまま送られてくるので、それに従って次の対応をおこなうだけです。つまり、パソコン通信というオートパイロットと大差ありません。CompuServe 用の CIM⁶でおこなっていることとレベル的には同じでしょう。両者の相違は、インターネットの広いコネクティビティと、分散リソースの管理、名前との関係づけだけだと言ってもよいほどです。

WWW の場合は HTML の記述と URL によるリソースへのタグで、そして Gopher の場合はメニューによって実現されています。

分散するリソースを関連づけ、結合していく方法によって、インターネットの応用はさらに広がっていくと思われます。もともと自律的な相互関係がなく、作成の動機もばらばらなドキュメントやアプリケーション、ソフトウェア、データベースを結合する試みがおこなわれています。WWW や Gopher にみられるこれらの試行は、

UNIX やインターネットで従来からおこなわれてきた階層的名前空間の管理の延長線上にあるということも重要な視点です。

インターネットの目指すものは、別個に生まれ、育ったものを結合して、さらに高度なサービスへ進化させていくとする姿勢です。これが、Mosaic によって垣間見えています。

Mosaic はたいへん大きなプログラムで、コンピュータのリソースを膨大に消費します。それゆえに、手許に高速なコンピュータを必要とする意義が実感できるのではないのでしょうか？ URL という表記法によって名前空間が統一的表现になったとはいっても、実際には別個のプロトコルが存在し、それらを全部解釈したうえで、ユーザーに対してある程度統一したインターフェイスを提供しなければなりません。これは並大抵のことではありません。これをハイパーテキストによって作られる空間のなかに組み込んでいくわけです。

もちろん、新たなプロトコルを提案して、そのなかへの統合を目指す方法もあるでしょう。しかしこれは、古いもの、あるいはすでに使っているもの、そして現在そのやりかたに馴染んで使っているユーザーに、場合によっては負担を強いることになります。ユーザーの手許に高性能なコンピュータがあるのですから、下には古いインターフェイスとの互換性を残したまま、上に高度なユーザー・インターフェイスを導入していくことは可能です。

たとえば、キャラクタベースで提供されるサービスに対して、GUI を構築することなども含まれるでしょう。このケースは、個々のサービスの提供するインターフェイスの個体差が大きすぎるためインターネットでは具体例が少ないのですが、CompuServe へのアクセスをおこなうためのプログラムなどにはこのようなものが見られます。

冒頭に紹介した NIFTY-Serve を Telnet でインターネットから利用するというのも、インターネットの目指すグローバルで連続的な知的環境を構築するステップの 1 つです。さらに相互にサービスを交換することで、より大きな環境に発展することができます。

今回は、Mosaic の多言語化版の動向を含め、Mosaic を使ってインターネットのリソースを探ることに挑んでみることにしましょう。

(よしむら・しん IJ)

6 CompuServe Information Manager: DOS、Windows のソフトウェア。Windows 版では、アイコンベースで CompuServe のサービスの Navigate ができる。

インターネットの
利用と仕組み

II

吉村 伸

Mosaic

はじめに

インターネットに対する注目が最近とみに大きくなっています。しかも、多くの場合はなはた魅力的に語られています。では、その魅力はどのようなところにあるのでしょうか。よく耳にするキーワードに、次のようなものがあります。

- 高速ネットワークである。
- タダである。
- 世界中とつながっている。
- 無料の情報が豊富である。

高速性に関してなら、米国では 45Mbps の NSFNET バックボーンをはじめとして、さらに高速なバックボーンの計画、インフォメーション・スーパーハイウェイ構想などにみられる部分が目を引きまします。たしかに、インターネットの技術はこのような高速ネットワークにも応用可能であり、現に動いているのも事実です。しかし、これについては機会を改めてとりあげたいと思います。

それでも、低速なネットワークから高速なネットワークにまで対応できる点には留意しましょう。ホモジニアスなネットワークは減び、ヘテロジニアスなネットワークが発展しているのです。

次の“タダ”という認識は誤りです。インターネットが無料で利用できるというのはまったくの誤解です。これは米国においても同様です。正しくは、“1人あたりのコストが安い”です。安価な技術が使えらることと、組織として接続してそのファシリティを共用できるためです。

どのようなワークステーション、あるいは PC にも TCP/IP が実装されるようになったため、機種にかかわらずインターネットに接続することができます。個々のコンピュータは、Ethernet などの安価なハードウェアを利用してインターネットに接続できるわけです。

インターネットは、組織内のローカルエリア・ネットワーク (LAN) から利用できます。広域インターネット接続のコストが高い我が国でも、LAN の構築にくらべると 1人あたりのコストは十分安くなります。これが、従来のインターネット・コネクティビティの姿です。したがって、大学では純粋に外部とのインターネット接続のために負担している金額は、構成員 1人あたりに換算すると年間数千円に満たないはずで

す。もちろん、これは大きな組織が享受できるメリットであって、インターネットが市民レベルに拡大するようになった最近では別の議論もおこなわれています。それが、米国ではもっぱら NII (National Information Infrastructure) をめぐってなされている議論です。テーマは、地域の公共機関、図書館や初等教育機関のインターネット接続、ケーブルテレビとインターネットの融合というふう

に発展しています。インターネットの技術は、ユーザー側からの要求に応えるかたちで形成され、きわめてオープンな場で論議されています。電子メールによる議論とオンラインの公開ドキュメントにより、費用もあまりかかりません。そしてなにより重要なのは、純粋な技術的論議によって決定され、新しい改良を柔軟にとりいれながらも素早く実行される点です。この部分が、国の代表が投票する国際標準とは大きく異なります。インターネットの発展に、インターネットの機能と利点が最大限に活用されているわけです。

しかも、世界規模のコネクティビティをもちながら、利用面ではきわめて柔軟です。プロフェッショナルは、IPによって確保されているインターネット上の接続性を利用したアプリケーションを自由に考えることができます。技術者に課される制約はきわめて少ないのです。

最後の無料の情報という点は、現にそうです。

インターネット上には、大学や公共機関などの提供する情報がたくさんあります。最近のインターネットの商業化を受けて、企業が提供する情報も増えてきました。しかし、伝統的にインターネットには USENET ニュースをはじめとするユーザー指向の情報が溢れています。

とはいうものの、ネットワークを流れる情報は膨大であり、質の点でも玉石混淆です。このような状況のなかからいかにして必要な情報を入手するかが、これからの時代に必要とされる能力かもしれません。

10 年ほど前に、“コンピュータ・リテラシー”という言葉がつかわれ始めました。リテラシーとは“読み書きの能力”という意味ですが、これは現時点では、コンピュータの使い方を知っているというより、“コンピュータとネットワークによって作られるサイバースペースでの暮らし方”というほうが妥当な解釈でしょう。

インターネットに情報が溢れていても、一方的に情報を受け取るばかりのブラックホールのユーザーが増えたらどうなるでしょう。インターネットの魅力が失われるのは明白ですが、そのようなユーザーはインターネットというサイバースペースのなかでどう評価されていくのでしょうか？

サイバースペースにおける人や企業に対する評価は、その考え、アクティビティをより多くの人が知ることによってなされていくでしょう。そのための場の形成は始まったばかりであり、今後の発展は未知数ですが、従来の“市場”に代わる概念が作られていくはずで、マルチメディア情報ブラウザ“Mosaic”の流行で、ユーザーにとっていかに有益な情報を提供するかひとつの尺度になりつつあるようにも思えます。

WWW や Gopher などによる情報の提供はごく簡単です。UNIX だけでなく、Macintosh や Windows NT などの PC もサーバーにすることができます。

今回は、“無料”の情報を効率よく見てまわるツール——とくに、別名“Internet Killer”といわれる Mo-

表 1 ftp.ncsa.uiuc.edu 配布のバイナリキット

Mosaic-alpha	DEC Alpha, OSF/1 version 1.3 No DTM, no HDF, no native WAIS
Mosaic-dec	DEC MIPS, ULTRIX 4.0 DTM, HDF, no native WAIS
Mosaic-hp700	HP 9000/730, HP-UX 9.01 No DTM, HDF, no native WAIS
Mosaic-ibm	IBM RS/6000, AIX 3.2.4, X11R5 DTM, HDF, native WAIS
Mosaic-sgi	SGI IRIX 4.0.x DTM, HDF, native WAIS
Mosaic-solaris	Sun Solaris 2.3 No DTM, no HDF, no native WAIS
Mosaic-sun	Sun SunOS 4.1.3 DTM, HDF, native WAIS

saic の使い方を中心に紹介します。Killer という名の由来は、使いやすくかつマルチメディア化されているため、大きなトラフィックを容易に惹き起こすからです。

WWW の事実上の標準クライアントとなっているのが Mosaic です。現在、X ウィンドウ・システム、Macintosh、Windows 上で動いています。使い方は共通です。以降では、X ウィンドウ・システム用の Mosaic を使って説明します。

Mosaicのインストール

Mosaic の入手

Mosaic の現在のバージョンは 2.1 です。X ウィンドウ・システム用はソースが公開されていますから、自分でコンパイルできます。これには Motif が必要ですが、Mosaic はバイナリも配布されているので、Motif がなくても使うことができます。

現在、表 1 のアーキテクチャに対応するバイナリが NCSA¹ から配布されています²。

このほかに、各ベンダーが配布しているものやボランティアに配布されているものがあります。これには、BSD/386 や FreeBSD/NetBSD、Linux 用があります。さらに、NCR、VMS、NeXT、UNIXware、DESQview/X³ などもあるようです。これらは国内の

1 ftp://ftp.ncsa.uiuc.edu/Mosaic/Mosaic-binaries

2 Motif を使って作ったプログラムのバイナリは自由に配布してもよいことになっています。

3 DOS 上で動作する X サーバーの 1 つ。

FTP サーバーにも置いてあります。前者の mirror は、www.ntt.jp や ftp.iij.ad.jp にあります。ftp.iij.ad.jp の BSD/386、FreeBSD のアーカイブにはこれらで動く Mosaic のバイナリがあります。NCSA で作成・提供されているバイナリ以外では、最新の 2.1 ではなく 2.0 のものもまだ多いようですが、とくに操作方法が異なるわけではありません。

X ウィンドウ・システム用の Mosaic は Motif を使っているために、残念ながら Motif のインプリメンテーションによる問題があります⁴。

表中の HDF (Hierarchical Data Format) と DTM (NCSA Data Transfer Mechanism) は、科学データ交換フォーマットの一種です⁵。

ネットワーク環境の確認

Mosaic では、ホスト名から IP アドレスが引けることが必要です。インターネットでは、DNS が引ける環境ということになります。したがって、その設定を確認する必要があります。

たとえば、Sun では NIS 経由で BIND を参照する設定が標準となっていますが、この設定でうまく動かないときにはリゾルバをリンクした Mosaic を使います。この場合、個々のマシンで `/etc/resolv.conf` にネームサーバーが正しく記述されていなければなりません。

この辺りの設定は各 OS によって異なります。マニュアルを確認し、管理者と相談して設定してください。

Viewer のインストール

Mosaic は、イメージデータを表示するときなどにいくつかの Viewer プログラムを起動します。イメージ・フォーマット、オーディオ・フォーマットの表示と再生をおこなうプログラムが、ハード・コーディングされています (表 2)。これらは一般的なプログラムですから、入手もインストールもとくに問題ないはずです。

Ghostview には Ghostscript が必要です。Ghostscript と xdvi は日本語化されたものがあるので、こちらをインストールしておくといでしょう。

4 たとえば Solaris2.x 用の Motif はサポートされておらず、バグが多くて使いものにならないなど。

5 興味のある方は、ニュースグループ `sci.data.formats` などを参照してください。

表 2 Mosaic から起動される Viewer

PostScript	ghostview
GIF、JPEG など	xv
dvi	xdvi
MPEG	mpeg-play
RFC822	metamail (mm)

オーディオデータの再生

Mosaic では音声データも扱います。最近のワークステーションや PC の多くは音声対応になっているので、たいていは音声データが扱えるはずです。

NCSA から配布されるバイナリの一部には、特有のツールを使うものもありますが、一般には showaudio が起動されます。

showaudio は、表 2 にある metamail (mm) のパッケージに含まれるシェル・スクリプトです。これを、個々の環境に合わせて修正します。自分が利用しているマシンに音声を再生する機能がないときは、別のマシンで音を出すように設定することもできます。

Xresources の設定

X ウィンドウ・システムのリソースの設定の大半は色やフォントに関するものなので、とりあえず動かすぶんにはデフォルトでも十分でしょう。しかし、最初に Mosaic を起動したときにアクセスするサーバーだけは設定してください。デフォルトでは www.ncsa.uiuc.edu になっていますが、これを最寄りのサーバーに変更します。もちろん、自分の組織のなかにサーバーがあればそれが最適です。この設定は次のように書きます。

```
Mosaic*homeDocument: http://www.ntt.jp/
```

このほかに変更したい部分があれば、一緒に配布されている `app-defaults` を参考にして修正します。

日本語、外国語表示

Mosaic-L10N⁶

日本語表示は、2.0、2.1 について NTT の高田さんらのおこなった L10N による多言語化によって実現しています。Motif 自体が日本語化されている必要はありませんが、すべての場所で日本語 (その他の言語) が使えるわ

6 Localization の略。

図 1 Mosaic-L10n による日本語表示

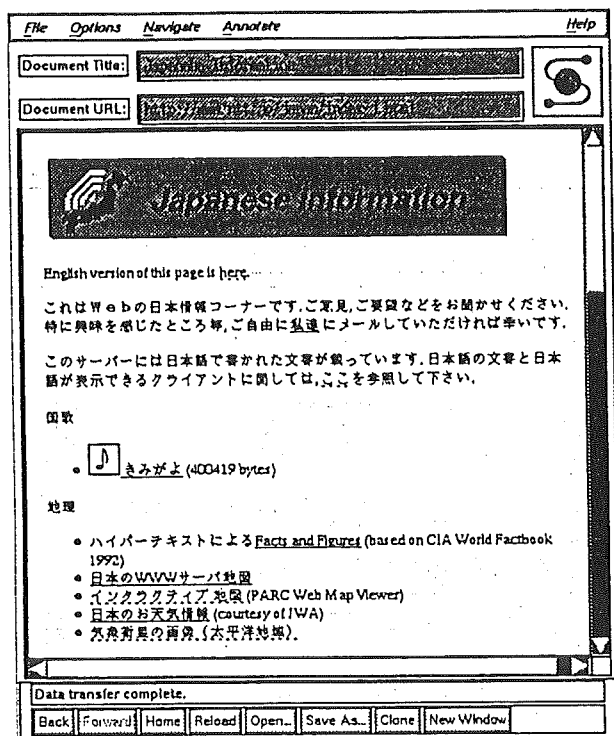
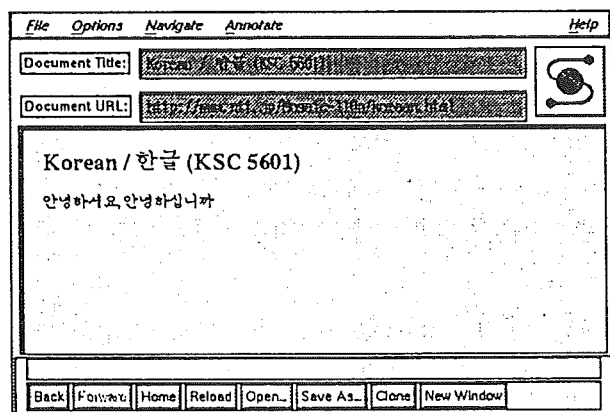


図 2 Mosaic-L10n によるハングル表示

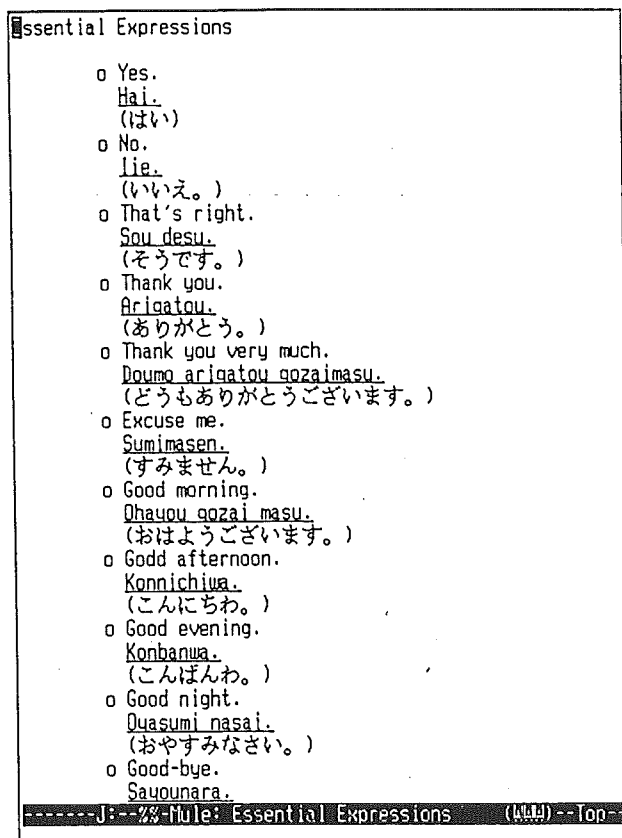


けではありません (図 1~2)。パッチは、www.ntt.jp、ftp.iij.ad.jp から入手できます。同じ場所に、いくつかのマシン用のバイナリもあります。

w3-mode on Mule

Mosaic-L10N では完全な意味での多言語化を実現しているわけではなく、現時点では 1 ページに複数の言語が混在しているケースには対応していません。この点については、GNU Emacs の多言語対応版 Mule を使うことによってカバーできます。Emacs Lisp で書かれた WWW インターフェイス w3-mode (図 3) を Mule で使うと、多言語化という点では完全なものになります。

図 3 Mule の w3-mode



Mosaic と比較すると、きれいな画面は出ませんが、そのぶん軽快に動きますし、テキストに関しては Emacs のなかで扱えるので、便利なこともあります。

Mosaic が事実上の標準になったため、多くの WWW クライアントは Mosaic との互換性を考慮しています。w3-mode も例外ではなく、Mosaic と同じ使い方ができるようになっています。

Mosaicの使い方

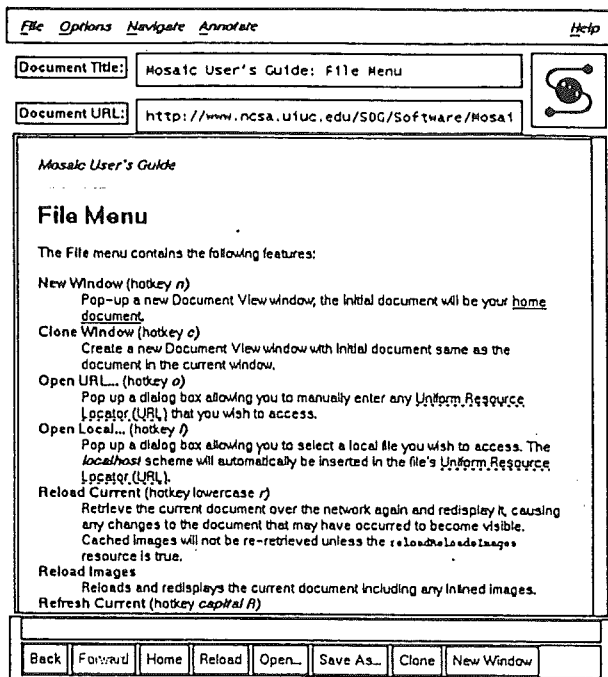
Mosaic の使い方を説明します。とはいっても、実際に使い始めればすぐに習得できるほど簡単です。

Mosaic の操作に関係するのは、最上段のプルダウン・メニュー、最下段のボタン、右上のグルグル回る地球の絵の 3 つの部分です。地球の絵のボタンは現在のアクセスを中止します。最初に憶えておきましょう。最下段のボタンは File メニューと、Navigate メニューのなかからよく使うものが選び出されています。

リソースの指定は、前回解説した URL (Uniform Resource Locators) によります。



図4 Fileメニューのマニュアル画面



Help

Helpメニューは、www.ncsa.uiuc.eduのサーバーへのアクセスになっています。たしかに、いかにもという感じです。Mosaicの使い方を、ハイパーテキスト・アクセス形式で見えていくことができます。

File

図4がFileメニューのマニュアル画面です。順に説明していきます。各項目の後ろの括弧内に書いた文字はホットキーで、これを入力すると機能を直接呼び出すことができます。

New Window (n)

新しいウィンドウを開く。

新たに開いたウィンドウは、ホームドキュメント(Mosaic*homeDocumentで指定されたリソース)を表示します。

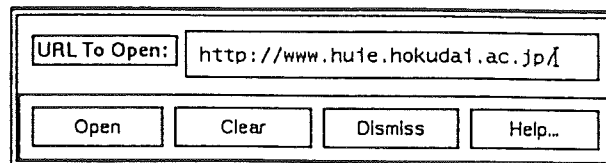
Clone Window (c)

現在表示しているものを複製して新しいウィンドウを開く。

Open URL (o)

URLを直接指定して、リソースへアクセスする(図5)。

図5 URLの直接指定によるアクセス



Open Local (l)

localhostのファイルを表示する。

HTMLによって記述されていれば、その指示による表示になります。

Reload Current (r)

現在表示している画面のドキュメントを再度アクセスして表示しなおす。

インラインのイメージは再アクセスしません⁷。

Reload Images

インラインのイメージを含めて再度アクセスし、描画しなおす。

Refresh Current (R)

画面を再描画する。

Reload Currentと違い、たんに描きなおすだけでネットワーク・アクセスは発生しません。

Find In Current (s)

現在の画面のなかで文字列を検索する。

View Source (d)

現在の画面のHTMLのソースを表示する。

Save As

現在のドキュメントをセーブする。

フォーマットは Plain Text、Formatted Text (ヘッダなどにわずかな飾りを付けたもの)¹、PostScript、HTML (生の Hypertext Transfer Management Language) のなかから選べます。

Print (p)

現在のドキュメントを印刷する。

フォーマットは Save As と同じく 4 つから選びます。

Mail To (m)

現在のドキュメントをメールで送る。

フォーマットは Save As と同じく 4 つから選びます。

Close Window (ESC)

現在のウィンドウを閉じる。

⁷ reloadReloadsImages: True になっていると、イメージも再アクセスします。

Exit Program

Mosaicを終了する。

Options

次は Option メニューです。これは、X ウィンドウ・システム固有の機能も含んでいます。

Fancy Selections

現在表示しているドキュメントを Cut & Paste するときのオプション。

これを on にすると、File メニューにおいて、Save、Print など Formatted Text を指定したときと同じようにペーストされます。Mosaic-L10N と kterm とのあいだでも問題なく Cut & Paste ができます。

Load to Local Disk

ハイパーリンクをマウスでクリックしたときに、ドキュメントを表示しないで直接ファイルに保存する。

ファイル名を入力するためのウィンドウがポップアップします。

Delay Image Loading

インライン・イメージを転送しない (図 6)。

日本のように遅いネットワーク上では有効です。インライン・イメージを転送して表示するには、次の Load Images in Current をクリックします。

Load Images in Current

未転送のインライン・イメージを転送して表示する (図 7)。

Reload Config Files

Mailcap や extension map を読みなおす。

Flush Image Cache

メモリ中のイメージ・キャッシュを消す。

Mosaic は一度読んだインライン・イメージデータはメモリ中にキャッシュしていて、Back、Forward ではイメージを読みなおしませんが、これを消してしまうと再度読みにいけます。メモリ、スワップなど計算機資源がとくに乏しい環境でなければ使う必要はないでしょう。

Clear Global History

リソースへのアクセス履歴を消去する。

Mosaic は、リソースへのアクセス履歴をホームディレクトリの .mosaic-global-history に保存していま

図 6 Delay Image Loading の実行

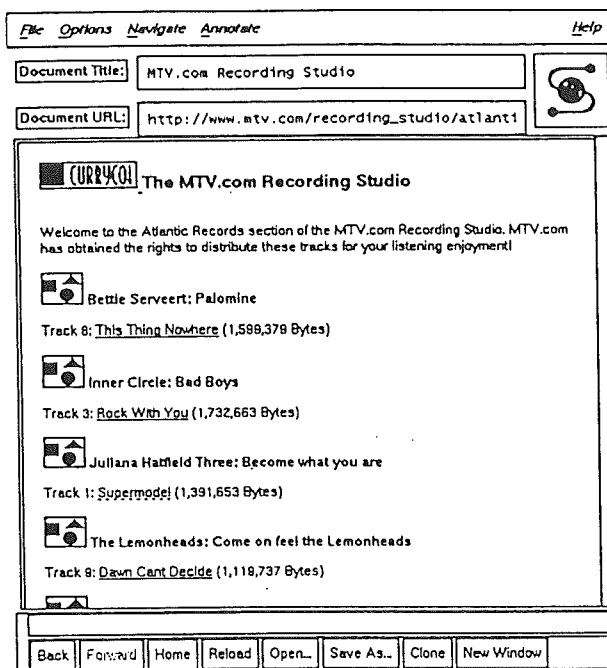
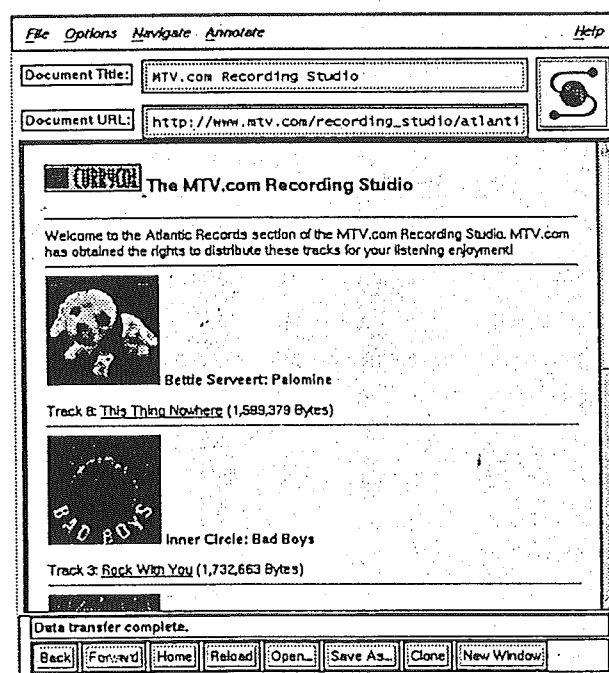


図 7 Load Image の実行結果



す。これと照合して、一度アクセスしたリソースは色やアンダーラインの形を変えて表示しています。

Fonts

フォント・ファミリーをその場で変更する。

Anchor Underlines

アンダーラインの形を変える。



Navigate

Navigate メニューは Mosaic を使ってインターネット・リソースを捜しまわるときに便利なものを集めたメニューです。

Back (b)

アクセス履歴を遡る。

アクセス履歴はウィンドウごとに作られていて、新たなウィンドウを作った場合は新たな履歴が開始されます。履歴はドキュメント・タイトルが表示され、タイトルと URL が保存されています。

Forward (f)

アクセス履歴を前に進む。

Home Document

ホームドキュメント (Mosaic*homeDocument に指示されているリソース) へ飛ぶ。

Window History (h)

アクセス履歴を表示し、そのなかの任意のドキュメントを表示する。

Hotlist

Hotlist を表示し、そのなかの任意のドキュメントにアクセスする。

次の Add Current To Hotlist で記録されたドキュメントにアクセスします。Hotlist はウィンドウにかかわらず共通で、過去の登録もホームディレクトリの .mosaic-hotlist-default に記録されています。

Add Current To Hotlist

現在表示中のドキュメントを Hotlist に加える。

Internet Starting Points

Internet Resources Meta-Index

多くのインターネット・リソースへのハイパーリンクを集めたもの。

www.ncsa.uiuc.edu の HTTP サーバーで提供されています。ユーザーは、URL を知らなくてもリンクをマウスでクリックするだけでアクセスできます。

Annotate

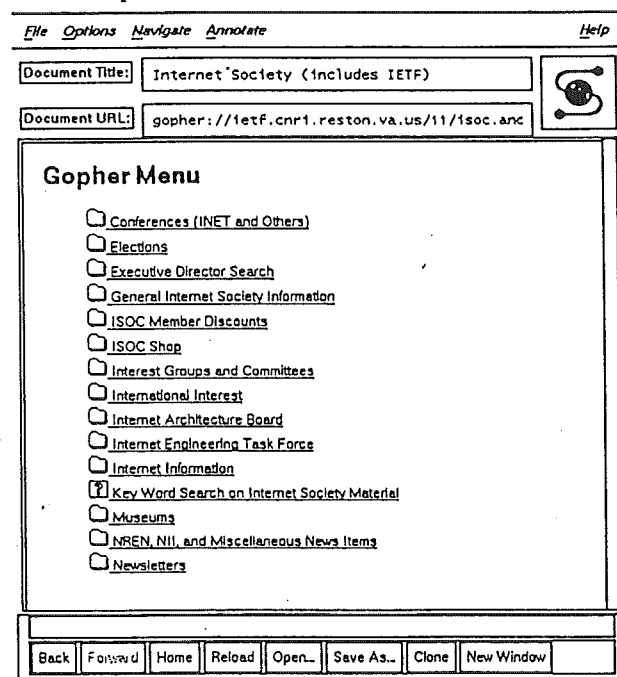
Annotate とは“注釈をつける”という意味です。アクセスしたドキュメントに自分の注釈をつけておくと、次にアクセスしたときには、そのドキュメントの末尾にあなたも追加されているかのように表示されます。

HTTP以外のリソースへのアクセス

Mosaic は、HTTP 以外にも URL で記述できる Gopher、FTP、news、Telnet でのアクセスができます。また、Mosaic-2.0 以降では、WAIS サーバーに直接アクセスできます。

Telnet は xterm を開き、目的とするサイトにアクセスします。

図 8 Gopher サーバーへのアクセス



Gopher サーバーへのアクセスは、ハイパーテキストではなくメニュー形式です (図 8)。フォントの切替えやインライン・イメージなどの派手さはありません。イメージ/音声データの表示/再生には、HTTP と同様に外部の Viewer が呼び出されます。

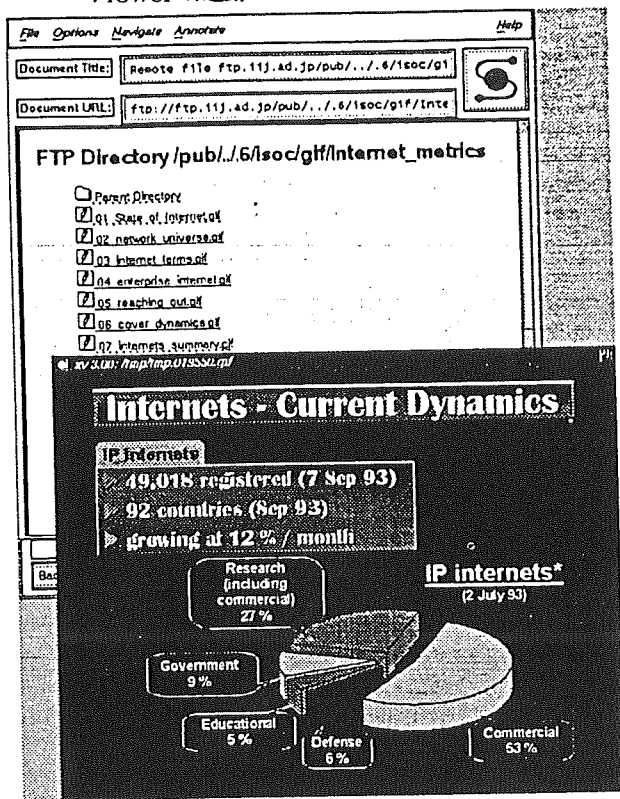
anonymous FTP、news、WAIS のそれぞれについて、例を紹介します。

anonymous FTP

ftp://<ftp site name>/

として、Open URL によってアクセスします。もちろん、ハイパーリンクで指定されてアクセスにいくこともあります。anonymous FTP では、ファイルの拡張子 (extension) に応じてアイコンが変化します。

図9 anonymous FTP サーバーへのアクセスと外部 Viewer の起動



アイコンは、ディレクトリはフォルダふう、バイナリデータは 01、02、……と番号が書いてあるもの、テキストはドキュメントふうに表示されます。アイコンは限られた属性判断しかされていませんが、.Z、.gz などの圧縮されているだけのファイルは、転送後に自動的に展開されて表示されます。gif、ps などの場合は、外部の Viewer が起動されます。tar.gz、tar.Z の場合はバイナリデータと判断され、ローカルファイルにします。

この判別は、ファイル名の拡張子だけをもとにした簡便なものです。したがって、Mosaic を FTP のクライアントとして使う場合、ターゲットのファイル内容がまったく分からないときは、Load To Local Disk オプションを指定して転送だけをおこない、表示しないようにするなどの工夫が必要です。

ファイル転送だけなら、わざわざ Mosaic で anonymous FTP をしなくてもよさそうなものです。しかし、anonymous FTP サーバーは依然としてもっとも鉱脈豊かなリソースです。ハイパーリンクの先にこれらを指定するなどして、compress や gzip で圧縮されているドキュメントを展開・表示したり、イメージを即座に表示できる

なら、インターネットのさらなる有効利用が図れるというものです。

news

NetNews、USENET ニュースへのアクセスは、最寄りの NNTPSERVER に対しておこなわれます。これには、環境変数 NNTPSERVER にサーバーが正しく指定されていることが必要です。

URL の指定では、ほかの場合と違ってサイトを指定しません。したがって、news:<newsgroup name>となります。

図10 USENET ニュースの表示

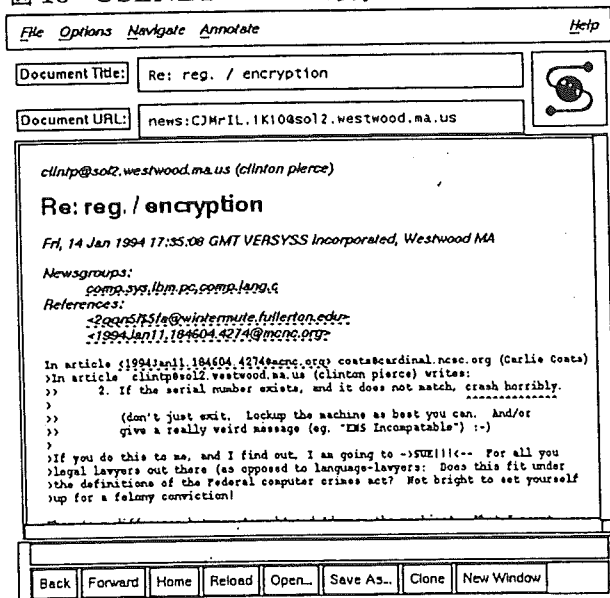


図10がその例ですが、Newsgroups と References の Message-ID がハイパーリンクになり、そこをクリックするとそのグループに飛んだり、そのメッセージを表示したりできます。Mosaic からのポストはできませんが、メールは出せるのでコメントを送ることは可能です。Message-ID に関しては、本文中に出てくるものもリンクになります。In article <...>のように書かれている部分も、そのメッセージを参照することができます。

WAIS

WAIS にアクセスするときは URL に、
wais://<wais server>/

と記述します。そのサーバーにどのようなソース (WAIS という 1 つのデータベース) があるかを知るには、



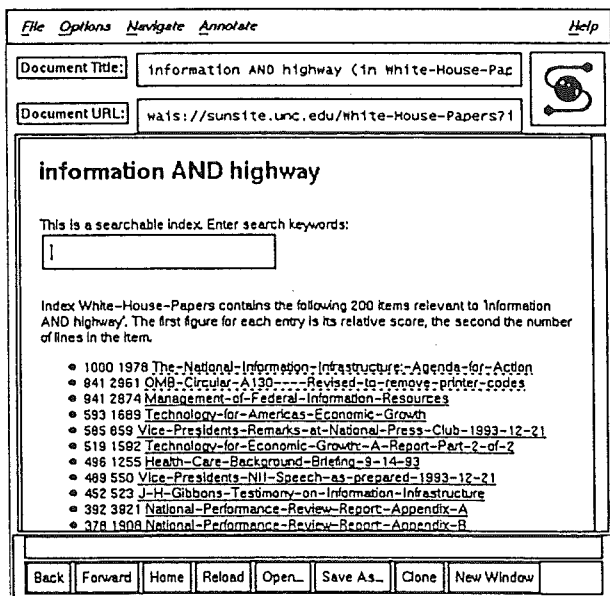
wais://<wais server>/?src とします。ちょっと乱暴な方法ですが、これは WAIS ではソースの拡張子が .src となっているからです。

たとえば、URL に、

wais://sunsite.unc.edu/White-House-Papers

と指定します (.src は付けません)。サーバーに接続すると、キーワードを入力するための窓が開きます。インターフェイスの面では、まだちょっと厄介かなという気がします。

図 11 WAIS のアクセス



おわりに

Internet Killer アプリケーション Mosaic には、Windows 用と Macintosh 用もあります。使い方はほぼ同じです。

Windows 用は WinSock 上で、Macintosh は Mac TCP 上で動きます。マルチメディアへの対応については、すこしずつ異なります。Windows や Mac が遅れているということではなく、Mac では QuickTime movie がサポートされているなど、プラットフォームに依存した相違です。

Mosaic を使っていると、サイバースペースを実感できるだけでなく、インターネットへの情報提供の大きな可能性を感じます。ハイパーテキストを核に、インターネット

のさまざまなプロトコルを利用して多彩な情報を扱い、まさにマルチメディアを駆使する情報提供ができます。

Mosaic はインターネットのコスト概念に対して一石を投じ、マルチメディア、高速ネットワーク化を促進する一翼を担っています。しかし、これは現時点で利用可能な仕組みで何ができるかという面から挑戦したアプリケーションです。問題がまったくないというわけでもなく、それらは今後議論されていくことでしょう。

しかし、情報提供の枠組として新たな一面を見せているのは間違いありません。

最近、ユーザーのニーズの多様化により、マスメディアによる情報流通、メディア流通が岐路に立たされているという話を耳にします。ネットワークの可能性との対比でも、さまざまな議論がなされているようです。

マルチメディアの方向性としては、On Demand TV の実現など、どちらかといえばエンターテインメント寄りの論議がかまびすしいようです。しかし、それによって情報を提供する側と利用する側の関係が崩れる可能性は低いでしょう。

これに対して、Mosaic は多様なニーズをユーザー間の要求の連鎖として捉え、それをブラウズするという新たな可能性を秘めています。

今後は、情報の享受だけではなく、“提供”面でのインターネットの“利用”についても積極的にとりあげていくつもりです。

(よしむら・しん IIJ)